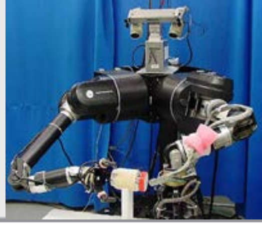
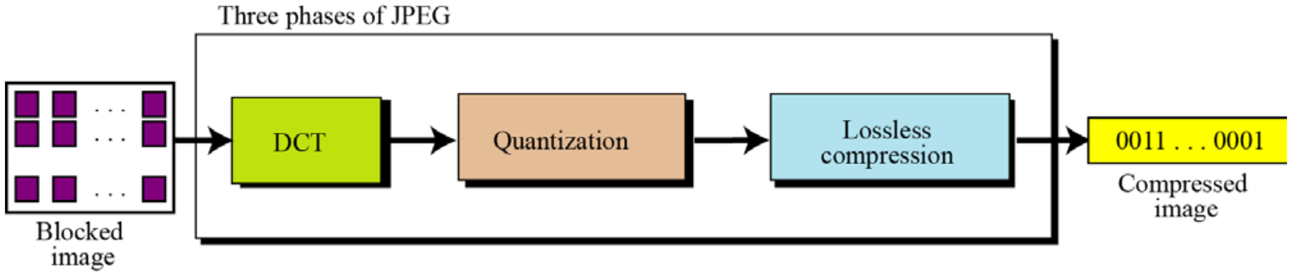
EVC - Computer Vision - Unit 17:
Image Encoding and Compression



- Content:
  - Introduction to Encoding
  - Image File Formats
  - Information vs. Data
  - Introduction into Compression
  - Lossless Compression
  - Lossy Compression
  - Video Compression



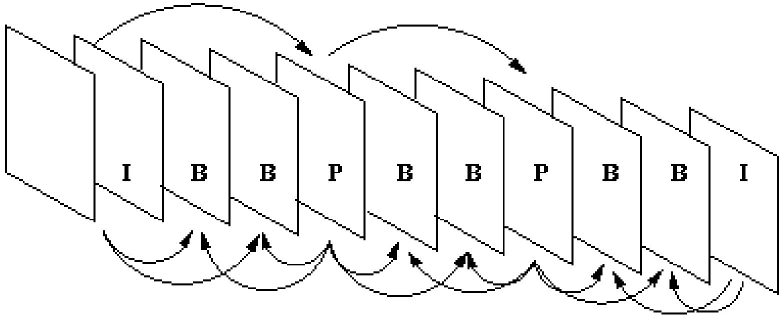
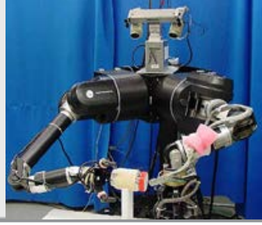


Figure 1: Prediction between MPEG–2 Frames

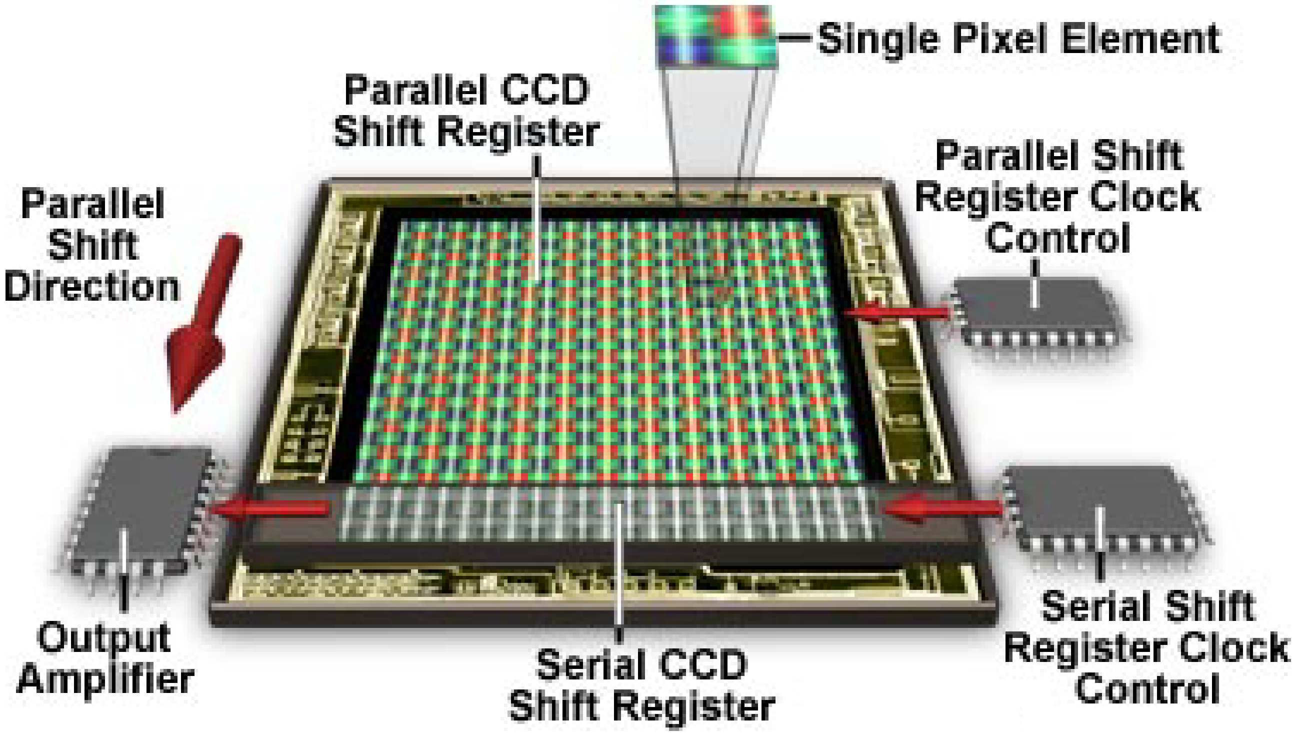
Encoding

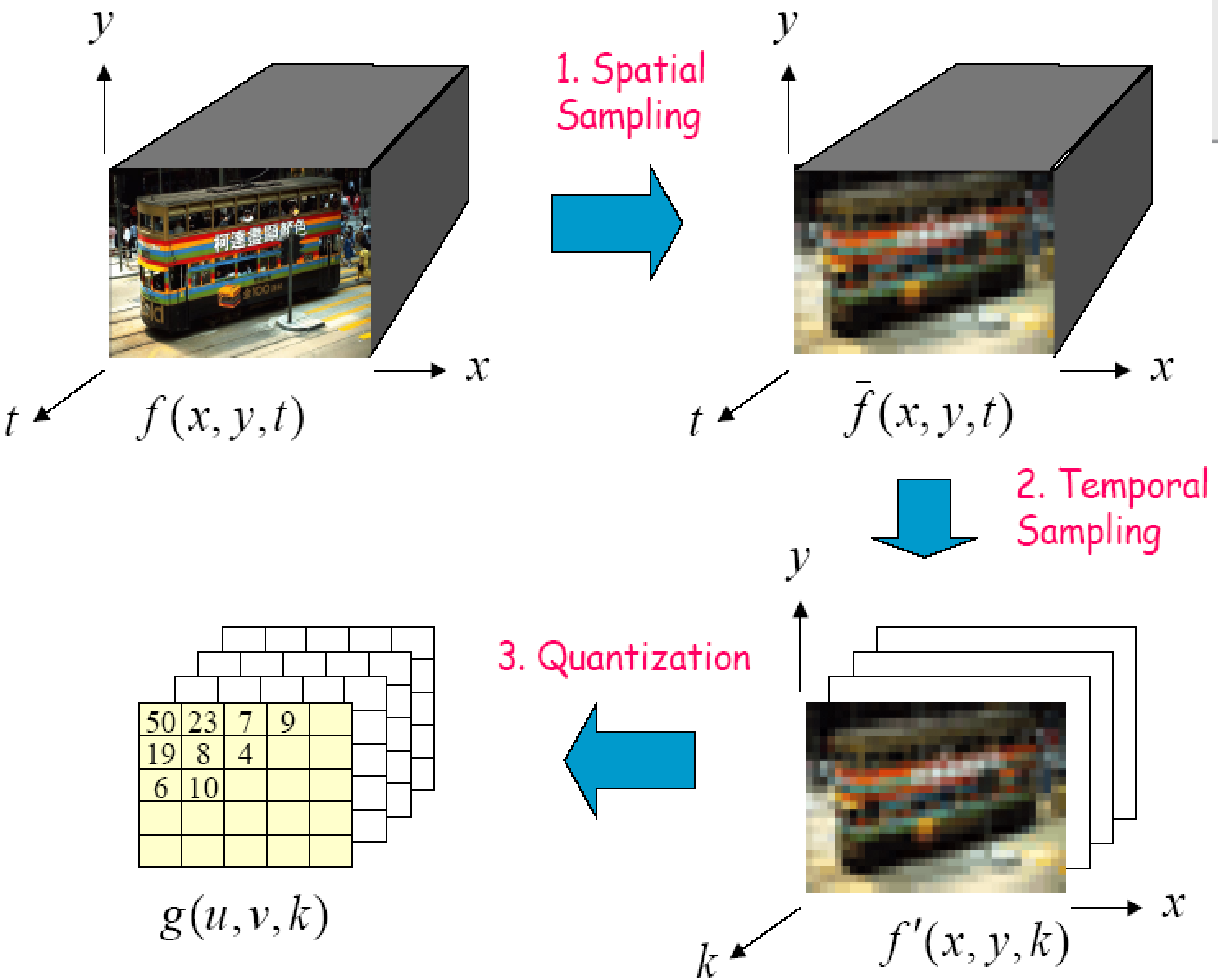
Image Acquisition using CCDs



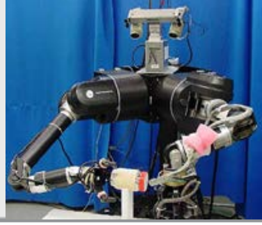
- Chip produces lines with analog values
- Fixed number of lines
- Lines are digitized
  - Space: Sampling
  - Intensity: Quantization
  - Time: Temporal Sampling
- Image Encoding
  - 2d matrix of digital values
  - File format?
  - Compression?

Full-Frame CCD Architecture

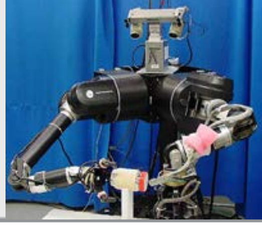

Single Pixel Element
Parallel CCD Shift Register
Parallel Shift Register Clock Control
Parallel Shift Direction
Output Amplifier
Serial CCD Shift Register
Serial Shift Register Clock Control



50	23	7	9	
19	8	4		
6	10			



Storage Requirements for Digital Images

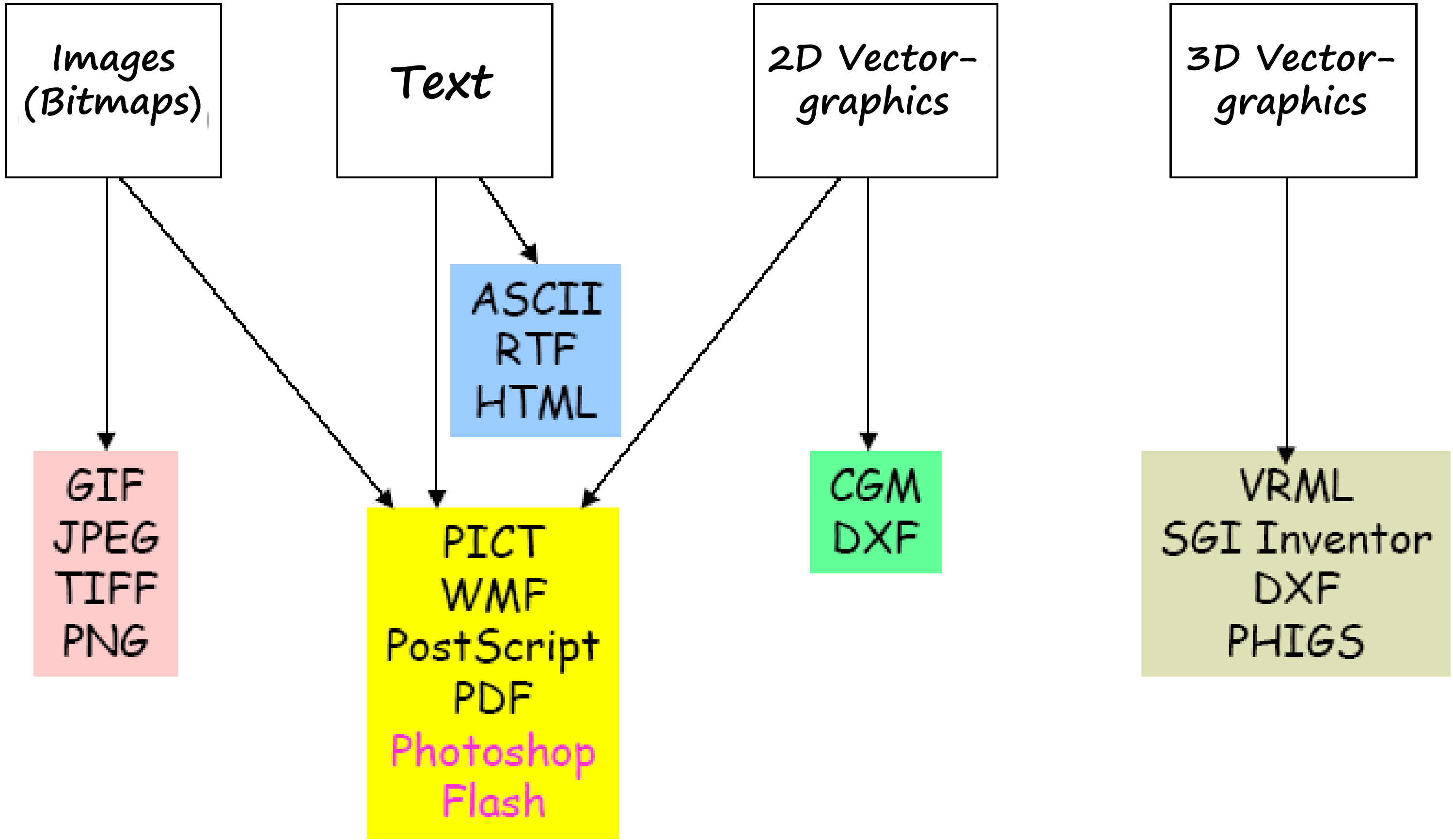


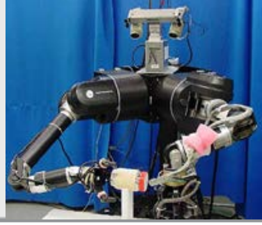
- Image *LxN* pixels, 2^B gray levels, *c* color components

Size = LxNxBxc

  - Example: *L*=*N*=512, *B*=8, *c*=1 (i.e., monochrome)
    Size = 2,097,152 bits (or 256 kByte)
  - Example: *LxN*=1024x1280, *B*=8, *c*=3 (24 bit RGB image)
    Size = 31,457,280 bits (or 3.75 MByte)
- Much less with (lossy) compression!

Image/Graphics Files





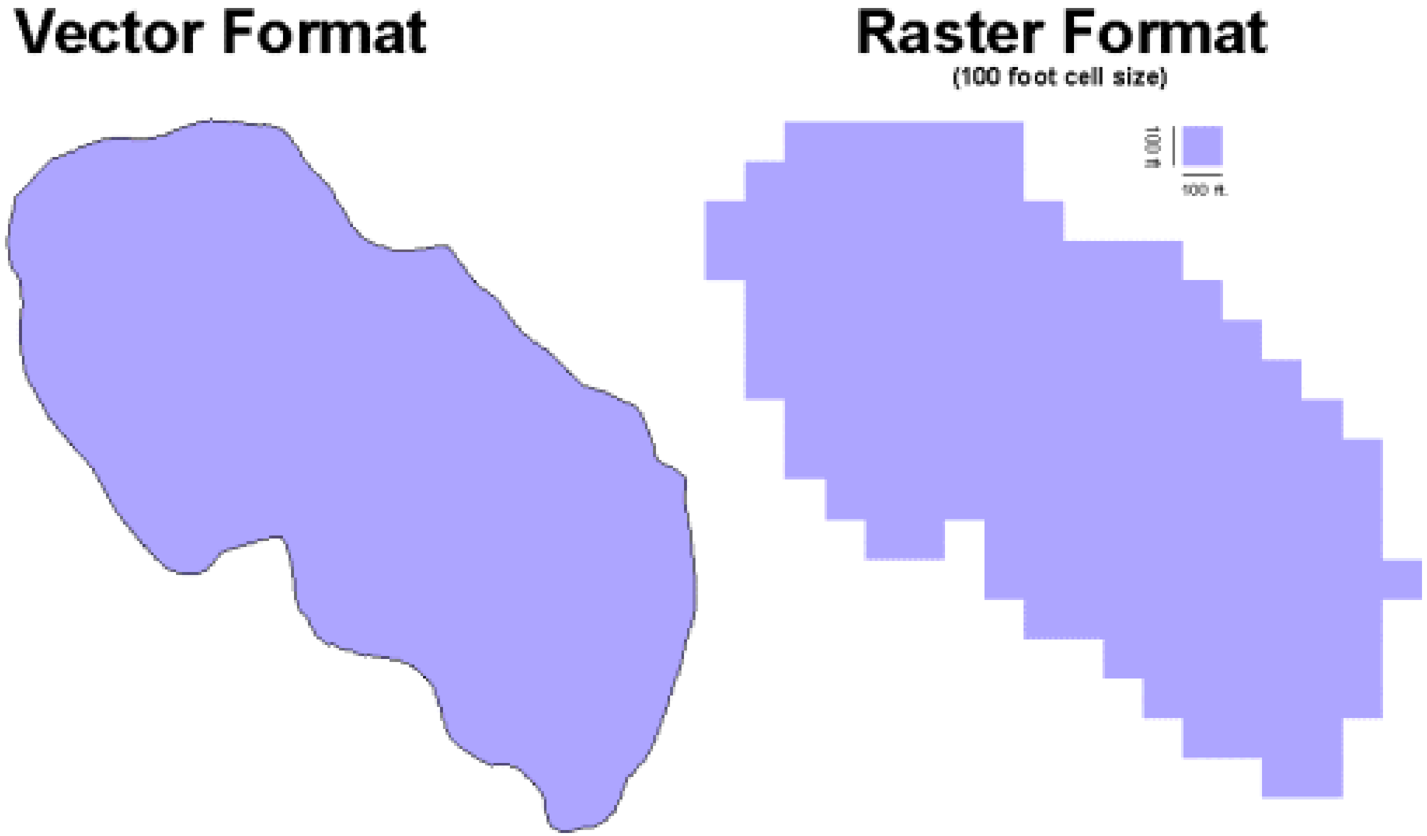
What are the Categories?

One categorization:

- Raster Image Formats
- Vector Image Formats

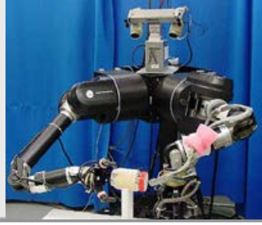
Another categorization:

- Binary Image Formats
- ASCII Image Formats



Raster Image Formats

Raster Image Formats



- Breaks the image into a series of color dots called “pixels”
- The number of bits at each pixel determines the maximum number of colors

1 bits = 2 (2^1) colors

2 bits = 4 (2^2) colors

4 bits = 16 (2^4) colors

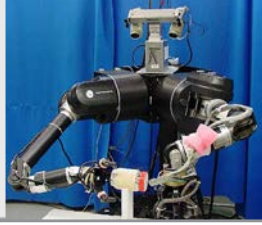
8 bits = 256 (2^8) colors

16 bits = 65,536 (2^{16}) colors

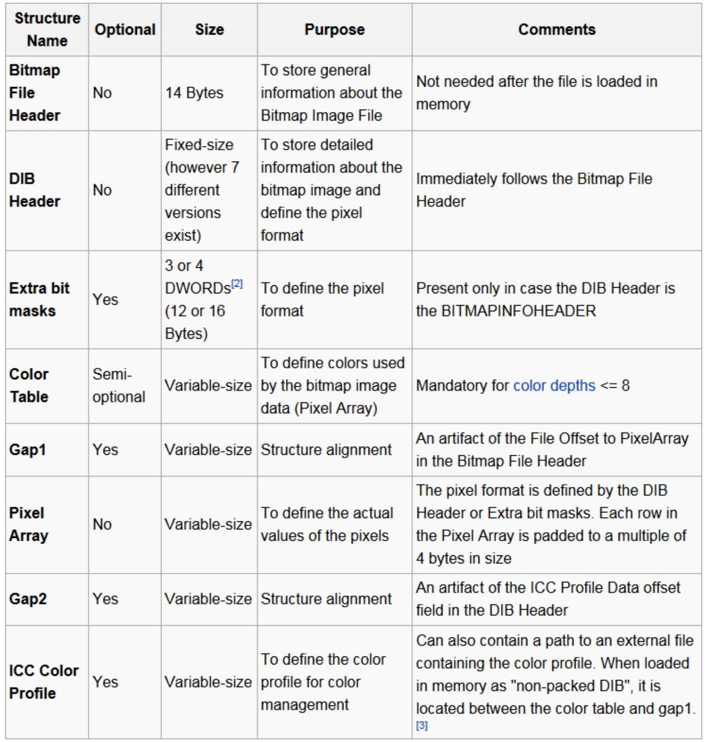
24 bits = 16,777,216 (2^{24}) colors

- Examples:
  - BMP/DIB: BitMaP or Device Independent Bitmap (DIB), Microsoft Windows and OS/2
  - PBM, PGM, PPM: Portable BitMap, GrayMap, PixMap, Unix, PC
  - TGA: Truevision Advanced Raster Graphics Adapter (TARGA), Avi

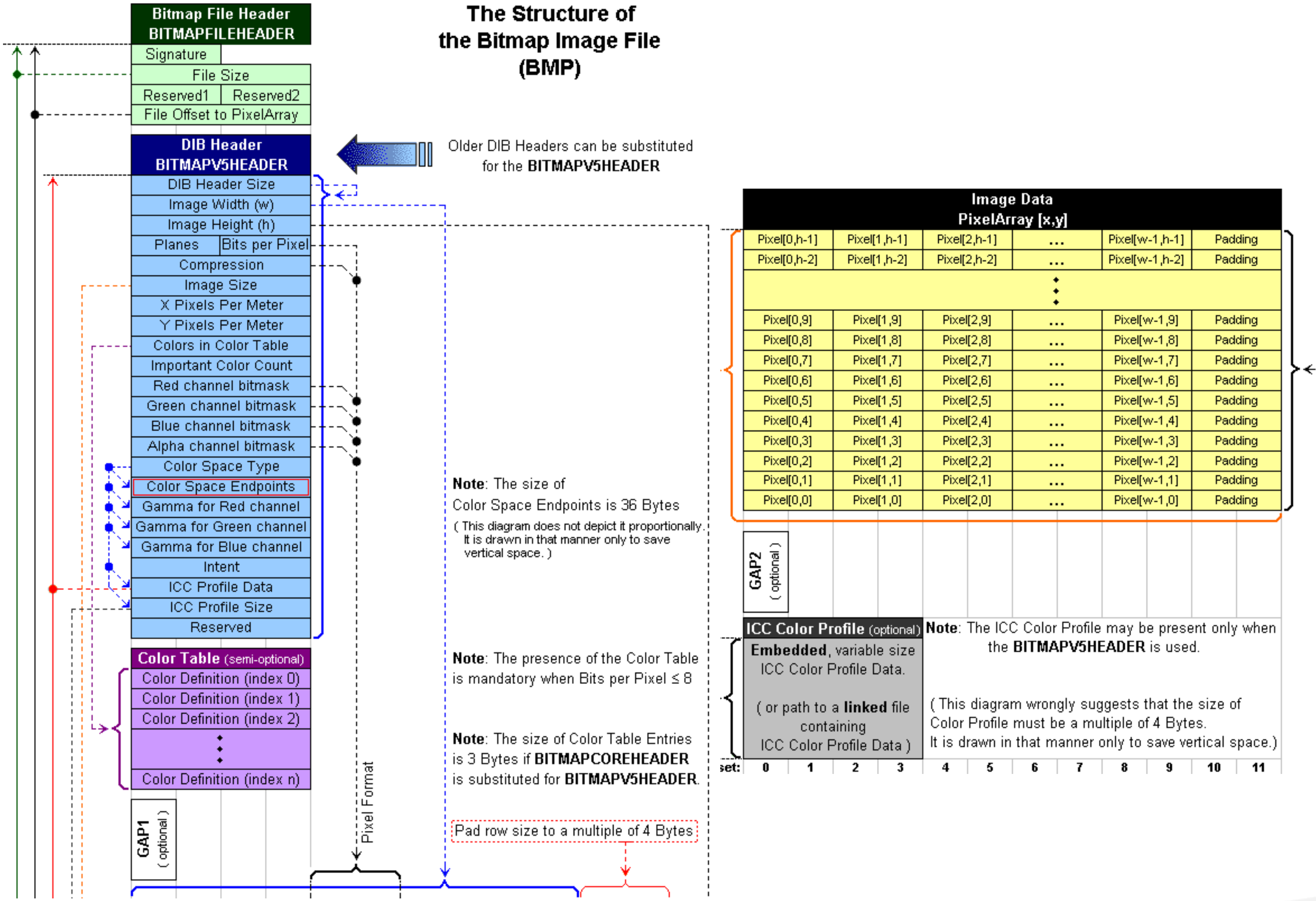
Example: BMP Format



- The bitmap image file consists
  - fixed-size structures (headers)
  - variable-size structures (image)



Structure Name	Optional	Size	Purpose	Comments
Bitmap File Header	No	14 Bytes	To store general information about the Bitmap Image File	Not needed after the file is loaded in memory
DIB Header	No	Fixed-size (however 7 different versions exist)	To store detailed information about the bitmap image and define the pixel format	Immediately follows the Bitmap File Header
Extra bit masks	Yes	3 or 4 DWORDs[2] (12 or 16 Bytes)	To define the pixel format	Present only in case the DIB Header is the BITMAPINFOHEADER
Color Table	Semi-optional	Variable-size	To define colors used by the bitmap image data (Pixel Array)	Mandatory for color depths <= 8
Gap1	Yes	Variable-size	Structure alignment	An artifact of the File Offset to PixelArray in the Bitmap File Header
Pixel Array	No	Variable-size	To define the actual values of the pixels	The pixel format is defined by the DIB Header or Extra bit masks. Each row in the Pixel Array is padded to a multiple of 4 bytes in size
Gap2	Yes	Variable-size	Structure alignment	An artifact of the ICC Profile Data offset field in the DIB Header
ICC Color Profile	Yes	Variable-size	To define the color profile for color management	Can also contain a path to an external file containing the color profile. When loaded in memory as "non-packed DIB", it is located between the color table and gap1.[3]



The Structure of the Bitmap Image File (BMP)

Bitmap File Header
BITMAPFILEHEADER
Signature
File Size
Reserved1 | Reserved2
File Offset to PixelArray

DIB Header
BITMAPV5HEADER
DIB Header Size
Image Width (w)
Image Height (h)
Planes | Bits per Pixel
Compression
Image Size
X Pixels Per Meter
Y Pixels Per Meter
Colors in Color Table
Important Color Count
Red channel bitmask
Green channel bitmask
Blue channel bitmask
Alpha channel bitmask
Color Space Type
Color Space Endpoints
Gamma for Red channel
Gamma for Green channel
Gamma for Blue channel
Intent
ICC Profile Data
ICC Profile Size
Reserved

Older DIB Headers can be substituted for the **BITMAPV5HEADER**

Color Table (semi-optional)
Color Definition (index 0)
Color Definition (index 1)
Color Definition (index 2)
⋮
Color Definition (index n)

GAP1 (optional)

Pixel Format

Note: The size of Color Space Endpoints is 36 Bytes
(This diagram does not depict it proportionally. It is drawn in that manner only to save vertical space.)

Note: The presence of the Color Table is mandatory when Bits per Pixel ≤ 8

Note: The size of Color Table Entries is 3 Bytes if **BITMAPCOREHEADER** is substituted for **BITMAPV5HEADER**.

Pad row size to a multiple of 4 Bytes

Image Data
PixelArray [x,y]

Pixel[0,h-1]	Pixel[1,h-1]	Pixel[2,h-1]	...	Pixel[w-1,h-1]	Padding
Pixel[0,h-2]	Pixel[1,h-2]	Pixel[2,h-2]	...	Pixel[w-1,h-2]	Padding
			⋮		
Pixel[0,9]	Pixel[1,9]	Pixel[2,9]	...	Pixel[w-1,9]	Padding
Pixel[0,8]	Pixel[1,8]	Pixel[2,8]	...	Pixel[w-1,8]	Padding
Pixel[0,7]	Pixel[1,7]	Pixel[2,7]	...	Pixel[w-1,7]	Padding
Pixel[0,6]	Pixel[1,6]	Pixel[2,6]	...	Pixel[w-1,6]	Padding
Pixel[0,5]	Pixel[1,5]	Pixel[2,5]	...	Pixel[w-1,5]	Padding
Pixel[0,4]	Pixel[1,4]	Pixel[2,4]	...	Pixel[w-1,4]	Padding
Pixel[0,3]	Pixel[1,3]	Pixel[2,3]	...	Pixel[w-1,3]	Padding
Pixel[0,2]	Pixel[1,2]	Pixel[2,2]	...	Pixel[w-1,2]	Padding
Pixel[0,1]	Pixel[1,1]	Pixel[2,1]	...	Pixel[w-1,1]	Padding
Pixel[0,0]	Pixel[1,0]	Pixel[2,0]	...	Pixel[w-1,0]	Padding

GAP2 (optional)

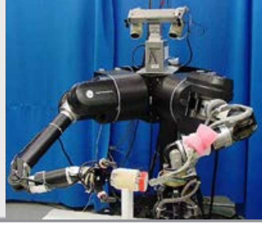
ICC Color Profile (optional)
Embedded, variable size ICC Color Profile Data.
(or path to a **linked** file containing ICC Color Profile Data)

Note: The ICC Color Profile may be present only when the **BITMAPV5HEADER** is used.

(This diagram wrongly suggests that the size of Color Profile must be a multiple of 4 Bytes. It is drawn in that manner only to save vertical space.)

set: 0 1 2 3 4 5 6 7 8 9 10 11

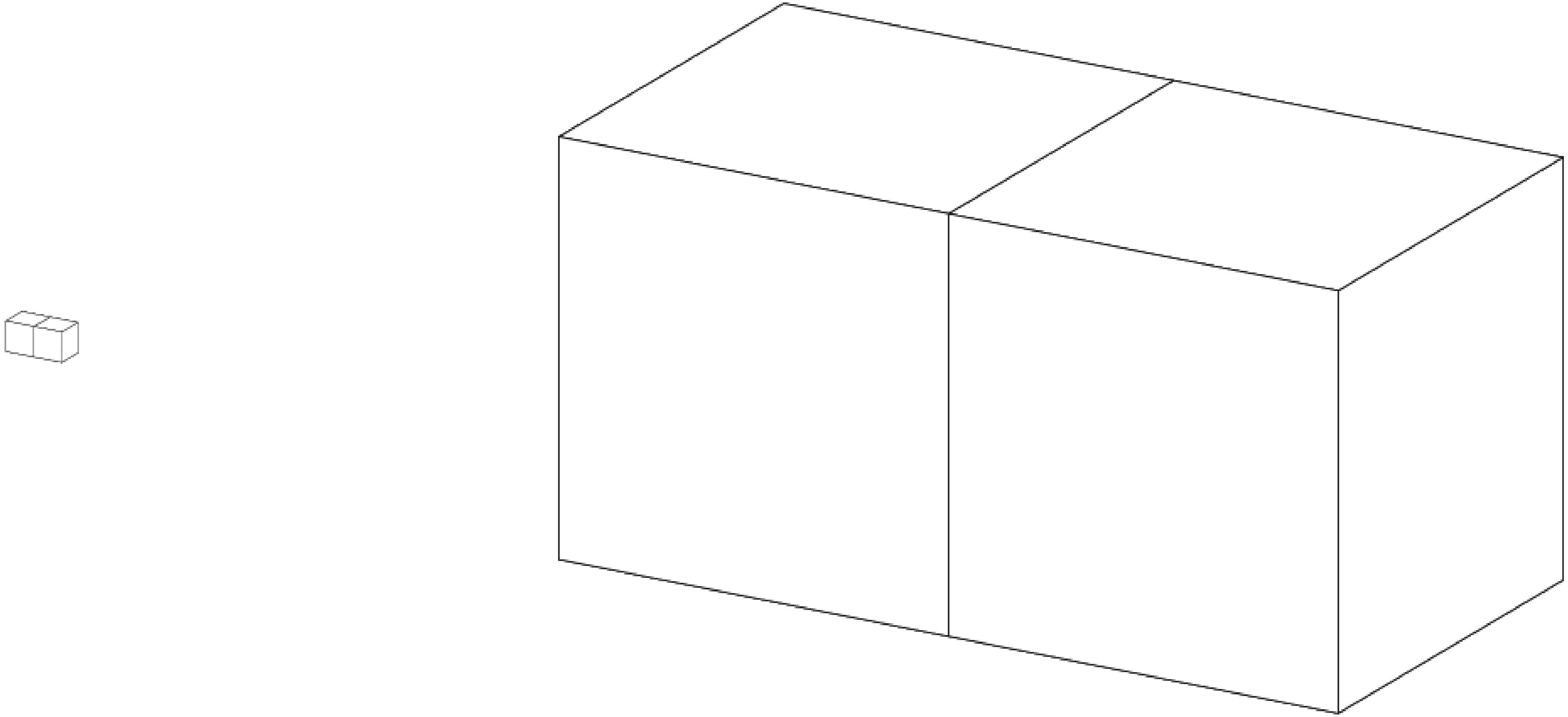
Raster Image Formats



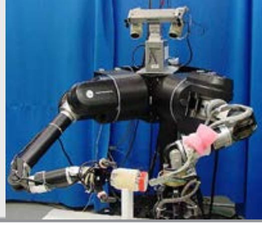
… more Raster File Formats

- PICT (Apple Quicktime))
- RAS Sun Raster Format (Sun Microsystems)
- RGB (Silicon Graphics)
- RLE Utah Raster Toolkit (Run-Length Encoded)
- XBM/XPM X-Windows Bitmap/Pixmap (X-Consortium), ASCII!
- GIF, TIFF, JPEG …

Instead …

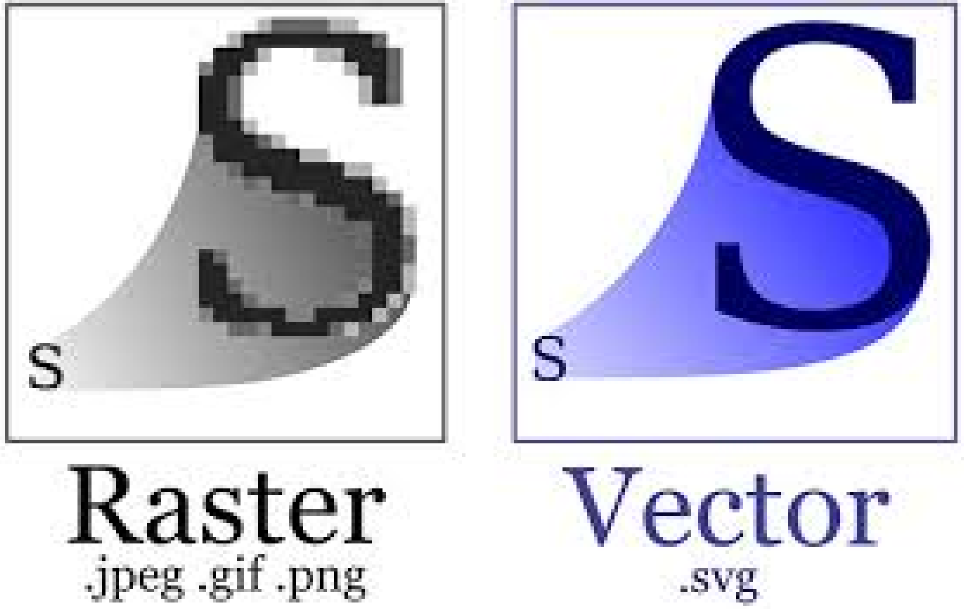


Vector Image Formats

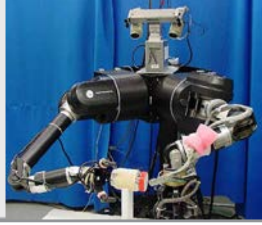


Vector Image Formats

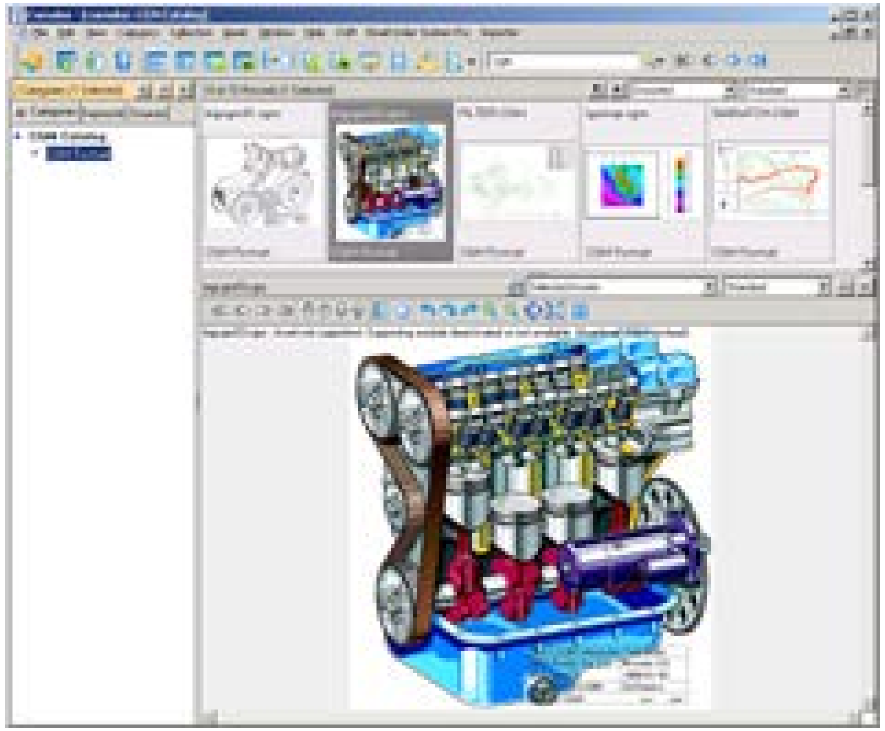
- Break the image into a set of mathematical descriptions of shapes: curve, arc, rectangle, sphere etc.
- Resolution-independent: scalable without the problem of “pixelating”.
- Not all images are easily described in a mathematical form.
- How to describe a photograph?



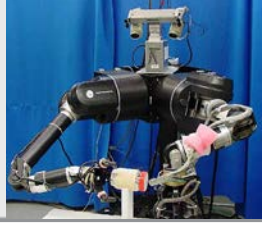
CGM



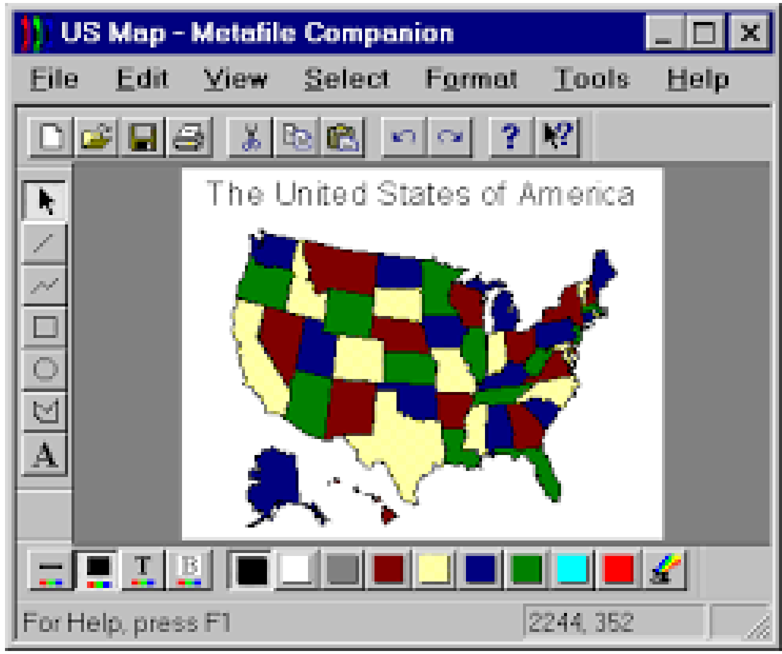
- Goal: to make vector graphics portable across different operating systems
- Computer Graphics Metafile: 3 types of coding
  - Raster / vector format, ANSI standard for exchange of image data between different graphics software (device independent). Metafile contains data and information, which describes the organization and the semantics of the data. Due to the structuring of CGM is an ideal partner for HTML and SGML.



WMF - Windows MetaFile



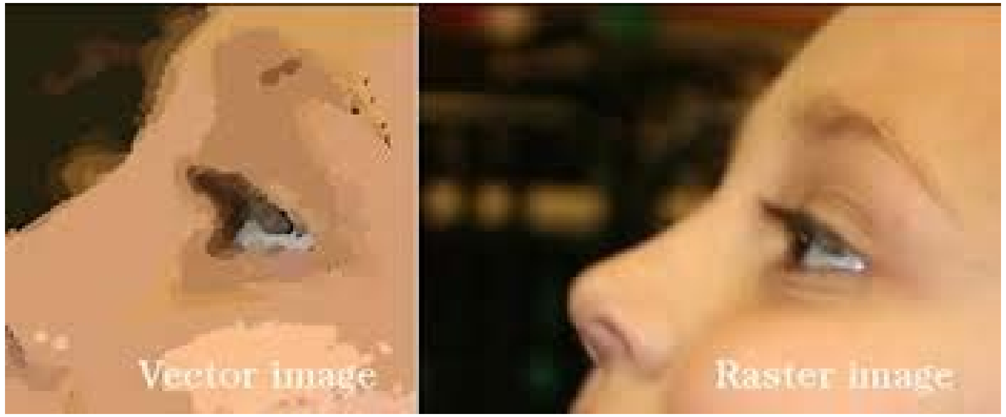
- Graphics file format on Microsoft Windows systems, originally designed in the 1990s. Windows Metafiles are intended to be portable between applications and may contain both vector graphics and bitmap components.
- WMF file stores a list of function calls that have to be issued to the Windows Graphics Device Interface (GDI) layer to display an image on screen.



Comparison

- Raster
  - Resolution-dependent
  - Suitable for photographs
  - Smooth tones and subtle details
  - Larger size

- Vector
  - Resolution-independent
  - Suitable for line drawings, CAD, logos
  - Smooth curves
  - Smaller size



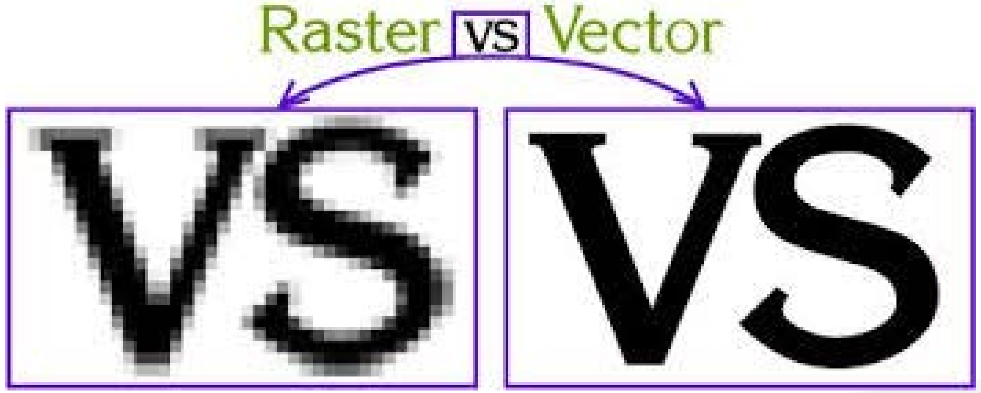
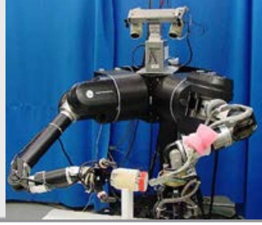
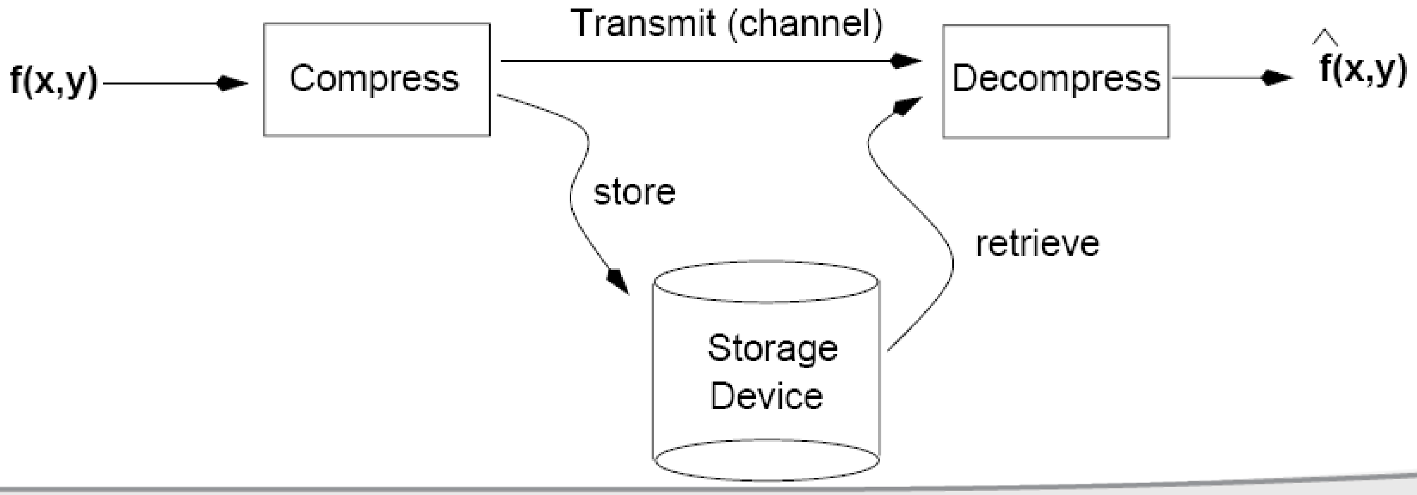


Image Compression

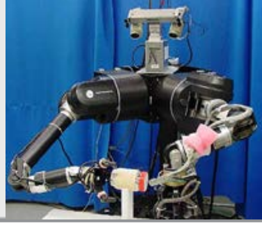
Goal of Image Compression



- Digital images require huge amounts of space for storage and large bandwidths for transmission.
  - A 640 x 480 color image requires close to 1MB of space.
- The goal of image compression is to reduce the amount of data required to represent a digital image.
  - Reduce storage requirements and increase transmission rates.



Data ≠ Information

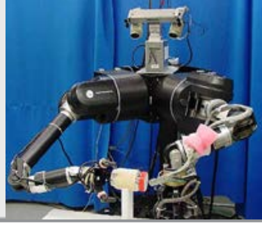


- Data and information are not synonymous terms!

- **Data** is the means by which **information** is conveyed.

- Data compression aims to reduce the amount of data required to represent a given quantity of information while preserving as much information as possible.

Data vs Information (cont’d)



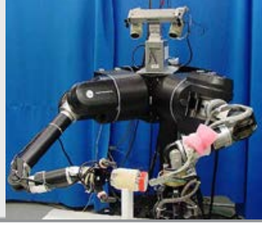
- The same amount of information can be represented by various amount of data, e.g.:

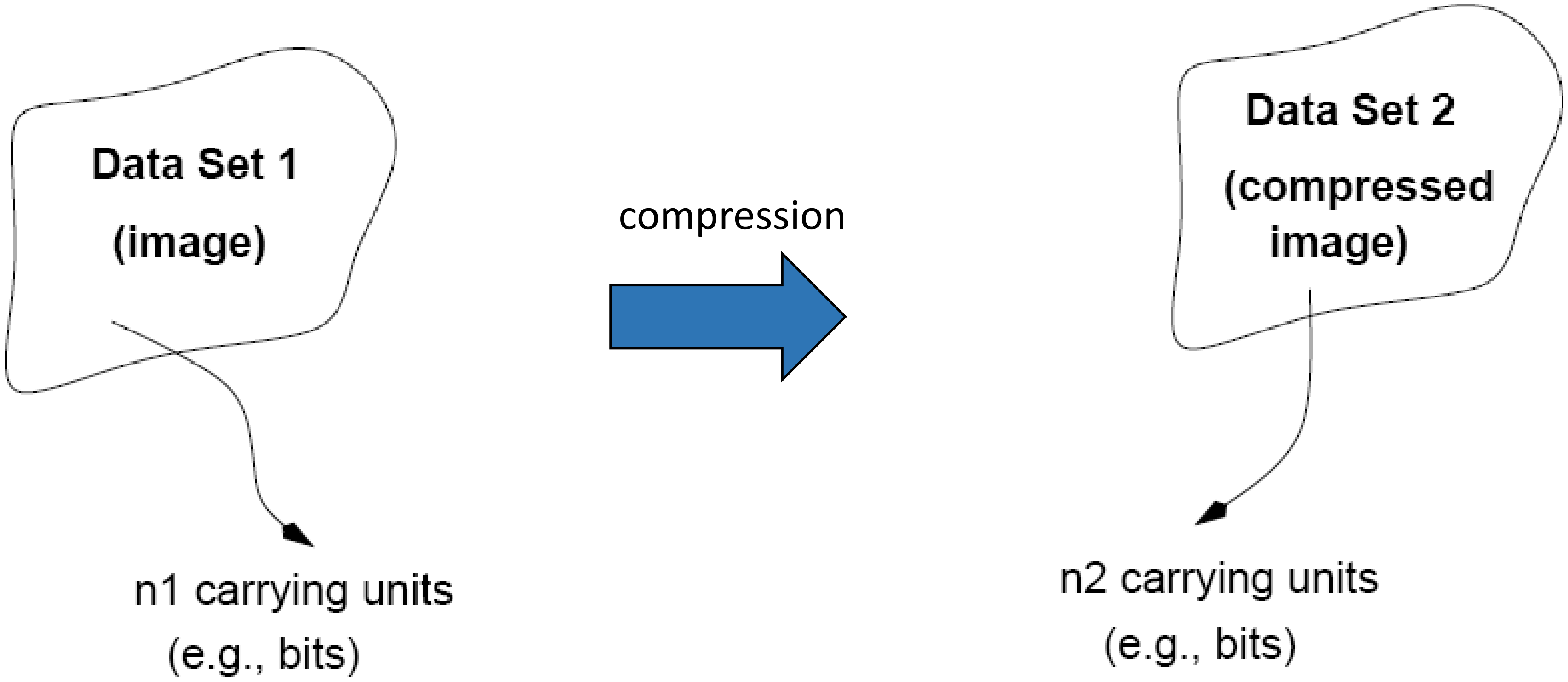
Ex1: *Your wife, Helen, will meet you at Logan Airport in Boston at 5 minutes past 6:00 pm tomorrow night*

Ex2: *Your wife will meet you at Logan Airport at 5 minutes past 6:00 pm tomorrow night*

Ex3: *Helen will meet you at Logan at 6:00 pm tomorrow night*

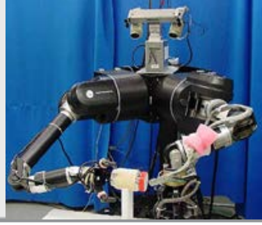
Data Redundancy



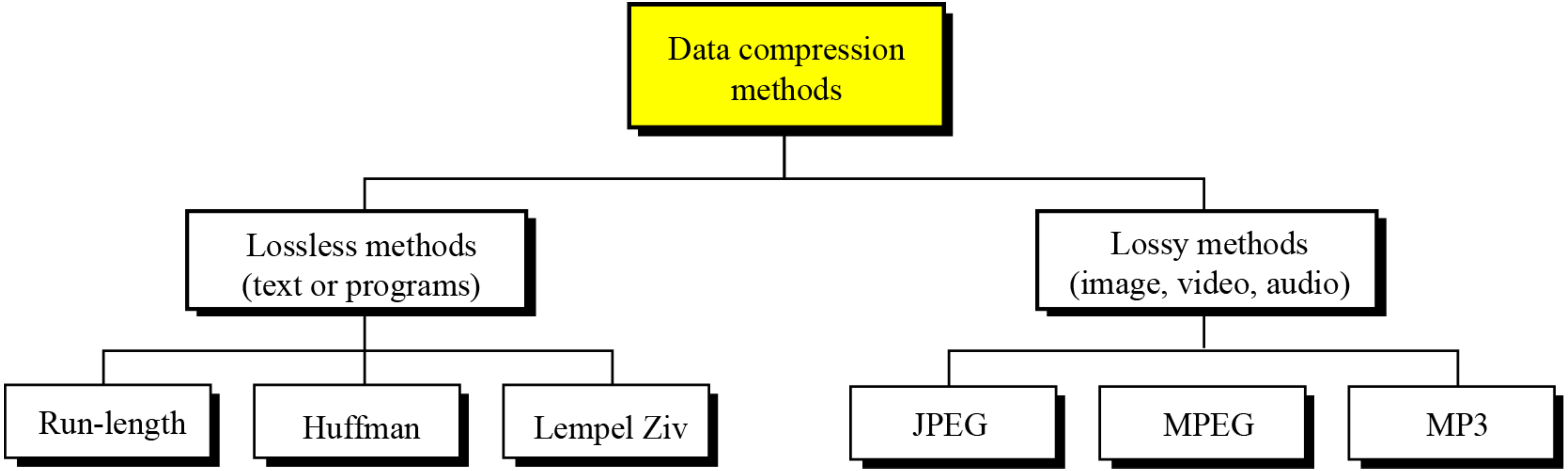


Compression ratio: $$C_R = \frac{n_1}{n_2}$$

Data Compression



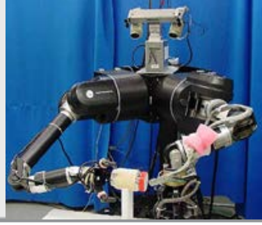
- Data compression implies sending or storing a smaller number of bits.
  - lossless and
  - lossy methods.
  - Trade-off: image quality **vs.** compression ratio



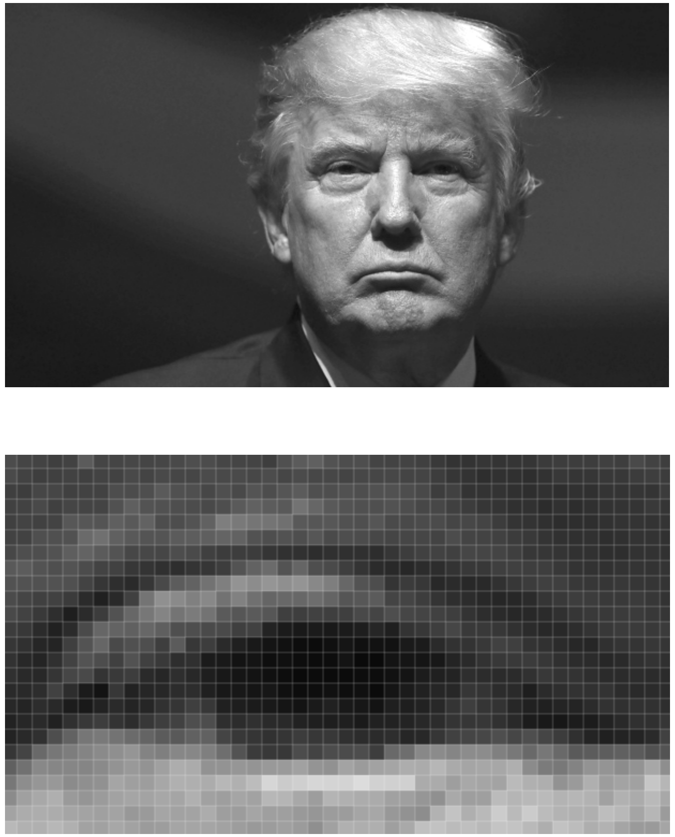
Data compression methods
Lossless methods (text or programs)
Lossy methods (image, video, audio)
Run-length
Huffman
Lempel Ziv
JPEG
MPEG
MP3

Lossless Image Compression

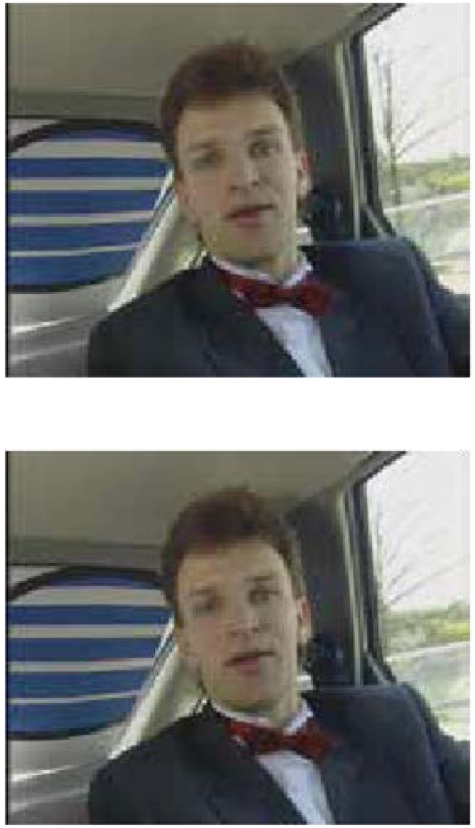
Run Length Encoding (RLE)



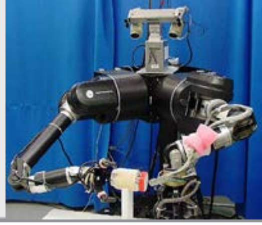
- Spatial and temporal neighboring pixels have similar intensity (colors)



spatial

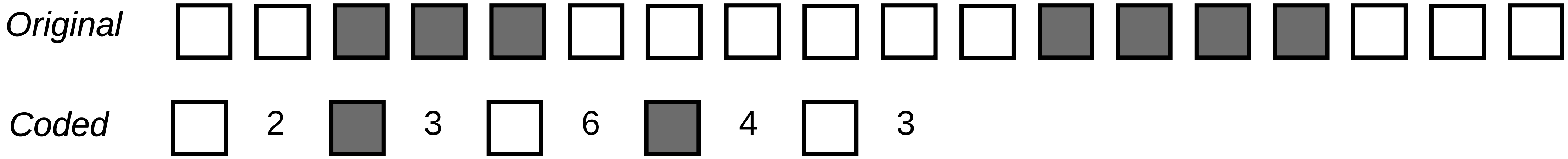


temporal



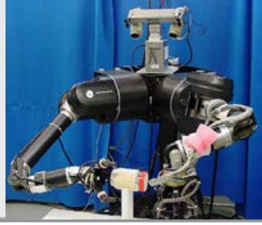
Run Length Encoding (RLE)

- Simplest method of compression
- Can be used to compress data made of any combination of symbols, does not need to know the frequency of occurrence of symbols
- Replace consecutive repeating occurrences of a symbol by one occurrence of the symbol followed by the number of occurrences



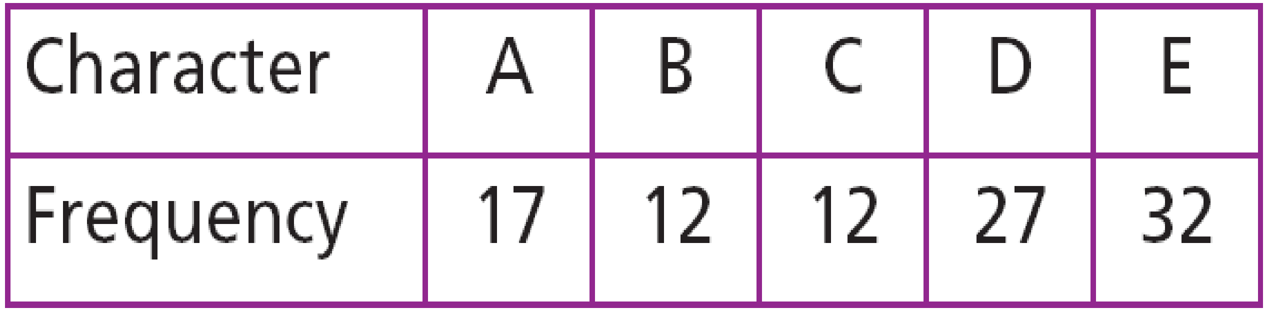
- Lossless compression!

Huffman Encoding



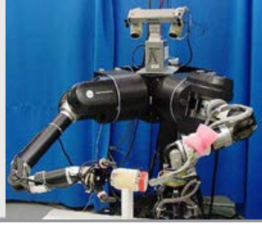
- Assigns shorter codes to symbols that occur more frequently and longer codes to those that occur less frequently.
- Example text file with five characters (A, B, C, D, E):
  - Assign each character a weight based on its frequency of use

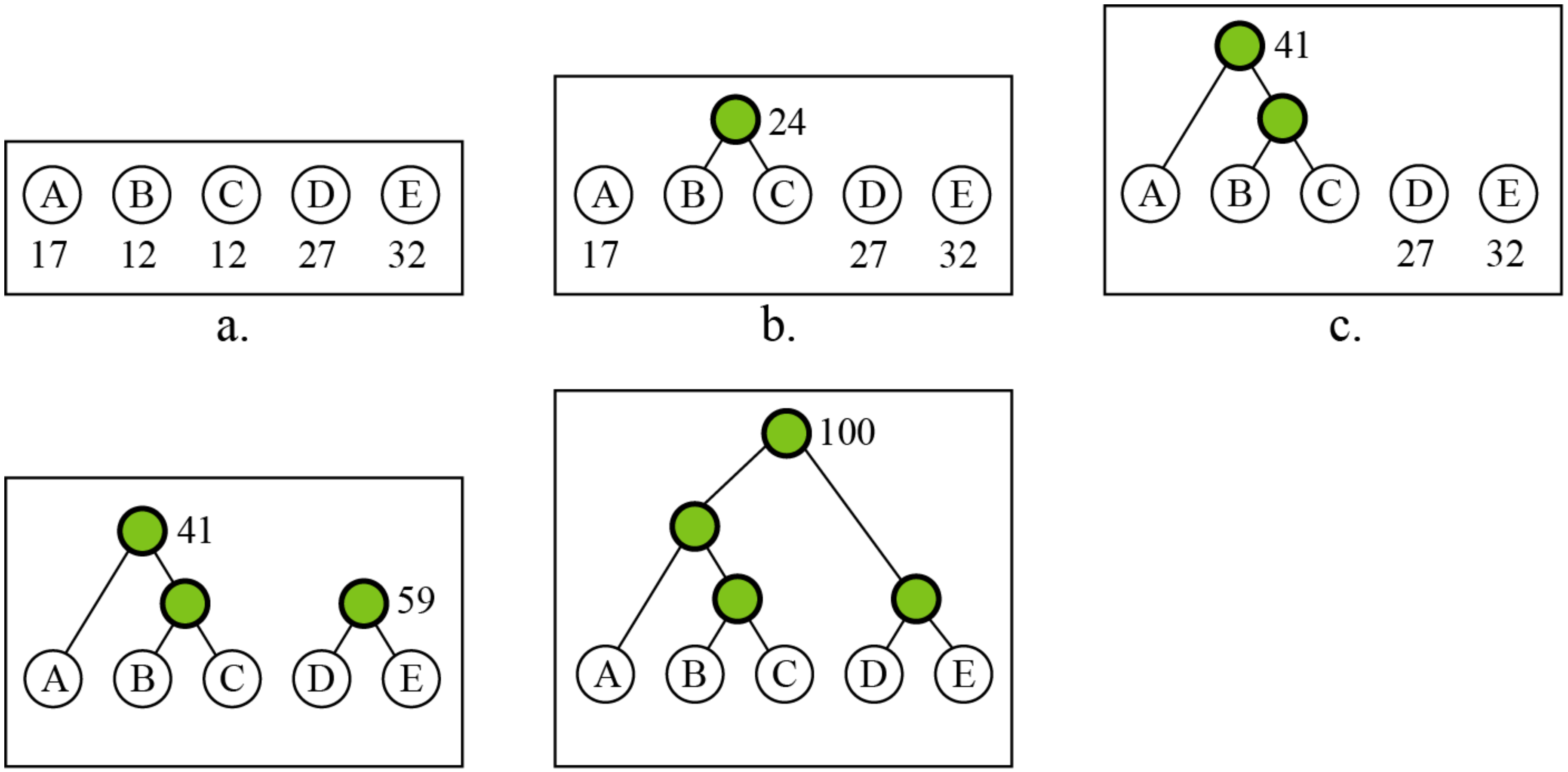
Frequency of characters



Character	A	B	C	D	E
Frequency	17	12	12	27	32

Huffman Encoding





a.

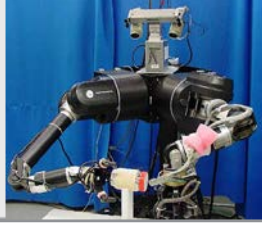
b.

c.

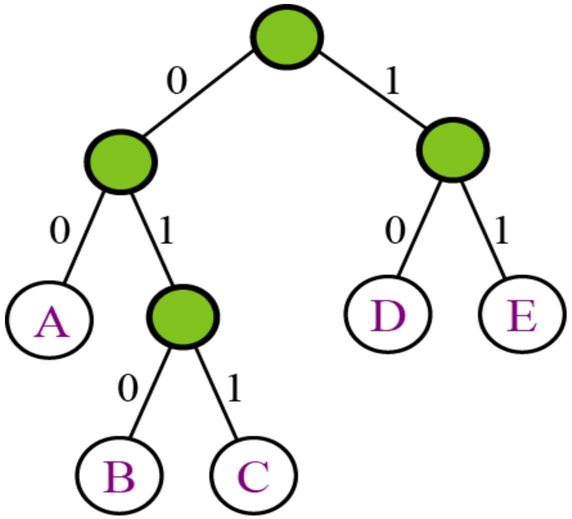
d.

e.

Huffman Encoding

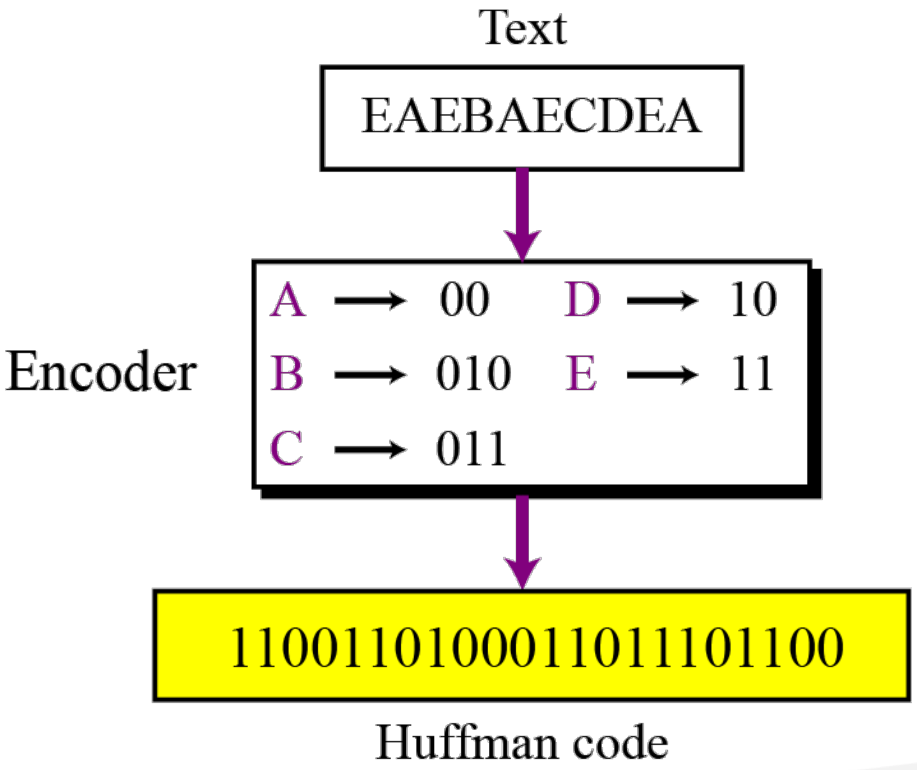


- Character code found by starting at the root and following the branches that lead to that character.
- The code itself is the bit value of each branch on the path, taken in sequence.



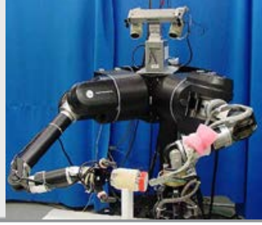
A: 00	D: 10
B: 010	E: 11
C: 011	

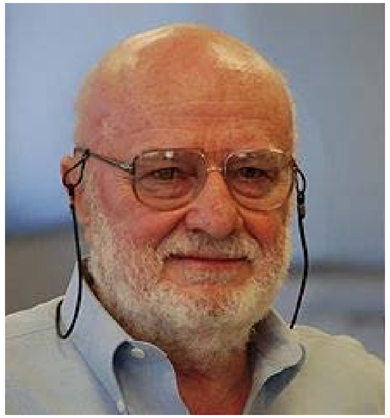
Code

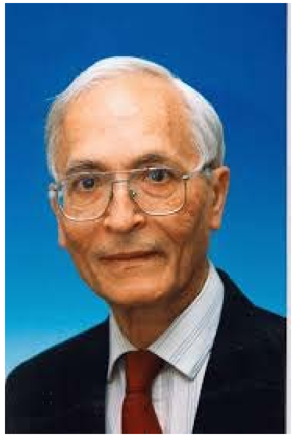


- Decoding: reverse process

Lempel Ziv (LZ) Dictionary-based Encoding

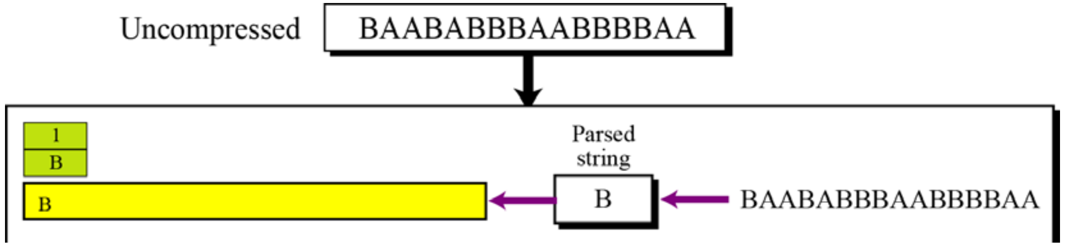


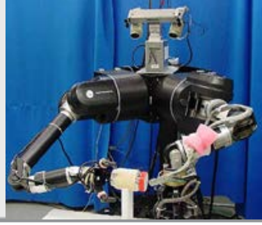




- Dictionary is a table of strings
  - Sender and receiver have a copy of dictionary
  - Previously-encountered strings are substituted by their index in dictionary
- Compression - two concurrent events:
  - Building an indexed dictionary
  - Compressing a string of symbols.
- Algorithm extracts smallest substring not in the dictionary from remaining uncompressed string.
- Stores a copy of this substring in dictionary as a new entry and assigns it an index value.
- Compression occurs when substring (except for the last character) is replaced with the index found in the dictionary.
- Process inserts the index and the last character of the substring into the compressed string.

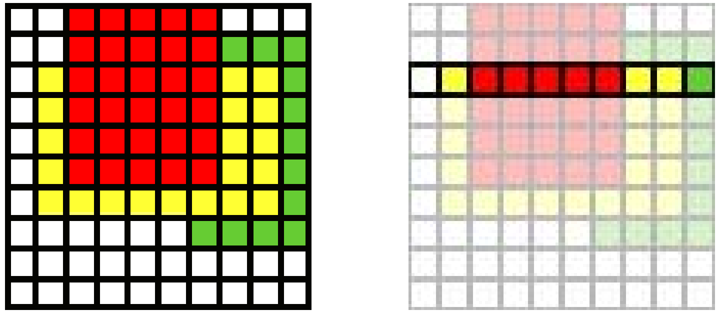
Example of Lempel Ziv Encoding



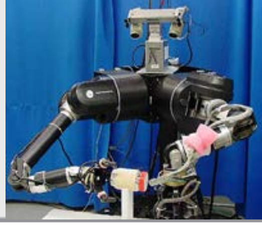


Lossless Image Formats

- GIF-Format (Graphics Interchange Format)
  - LZW-Compression (Lempel, Ziv, Welch): works line-wise



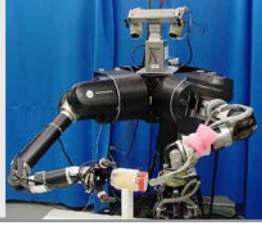
  - 1 st line, 2nd line
  - 3rd line is compressed as:
    - 1 white, 1 yellow, 5 red, 2 yellow, 1 green
  - Row 4 to 6: „as row 3“
- Indexed (1-8 bit Color Lookup Table)



CompuServ GIF (gif)

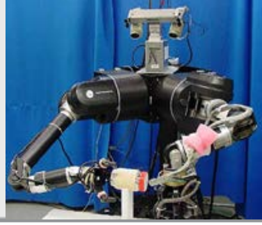
- First standardized in 1987 by CompuServ (called GIF87a)
- Updated in 1989 to include transparency, interlacing, and animation (called GIF89a)
- Use the LZW (Lempel-Ziv Welch) algorithm for compression (not free, licenses necessary)
- A maximum of 256 colors freely selectable out of 24 bit (2^{24} = 16.777.216)
- Does not work for photographs
- Suitable for small images such as icons
- Simple animations

Portable Network Graphics (png)



- Developed because of licensed LZW with gif
- Replacing GIF and TIFF (not JPEG!)
- 3 Types
  - True Color (3 x 16 bit/pixel)
  - Grayscale (1 x 8 bit/pixel)
  - Palette (256 colors)
- Compression similar to PKZIP (Phil Katz)

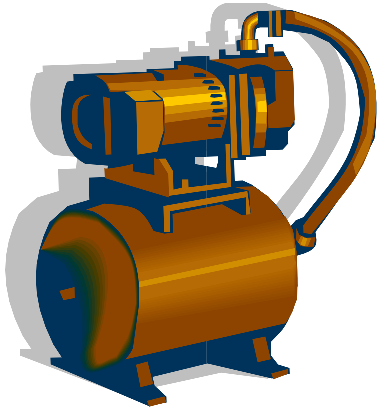
Tagged Image File Format (tiff)



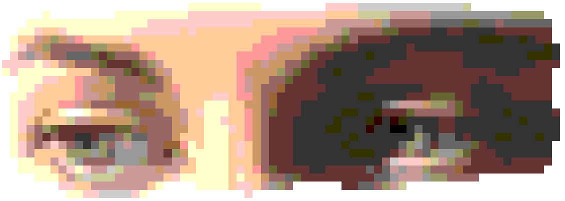
- Originally created as an attempt to get desktop scanner vendors of 1985 to agree on a common scanned image file format, rather than have each company promote its own proprietary format
- Aldus/Microsoft/Adobe
- Flexible and extendible because of „tags“
- Indexed or True-color
- Compressed/uncompressed (Raw)
- No standardization, more than 50 different formats
- JPEG compression

Lossy Image Compression

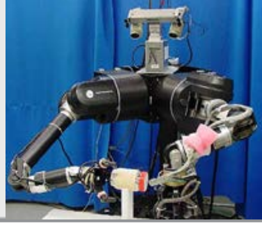
Lossy Compression



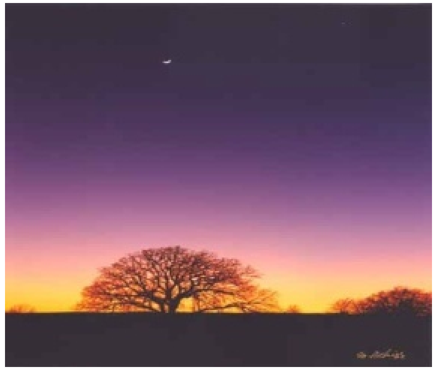
- Run-length coding works well for simple images (graphics) and computer data
- However, a large, detailed picture usually is not reduced enough
- In order to reduce information further lossy compression is needed
  - Information is permanently removed
  - The trick is to remove details that are not perceived by a human observer
  - Many of these psycho-visual coding systems take advantage of a number of aspects of the human visual system



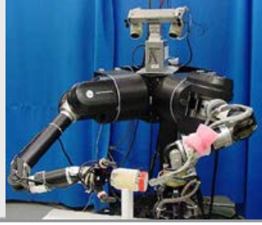
Human Visual System



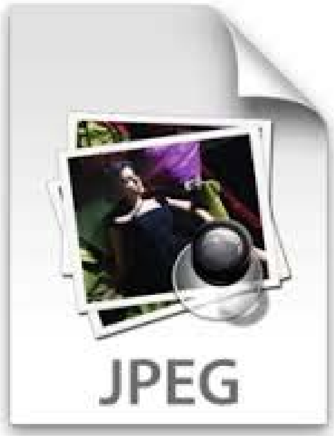
- The eye is more sensitive to brightness changes than to color changes
- The eye is not able to perceive brightness above or below certain threshold values
- The eye does not perceive little brightness or color changes. The strength of this phenomenon is dependent on the color



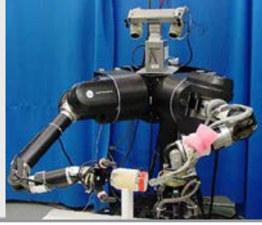
Compression of Still Images



- Characteristics of human vision have been transferred to lossy image compression techniques
- Such a development is the JPEG format
  - Stands for "Joint Photographic Experts Group“
  - JPEG is the worldwide standard
  - JPEG compresses brightness and color information separately
  - Color information is more compressed (lower sensitivity)

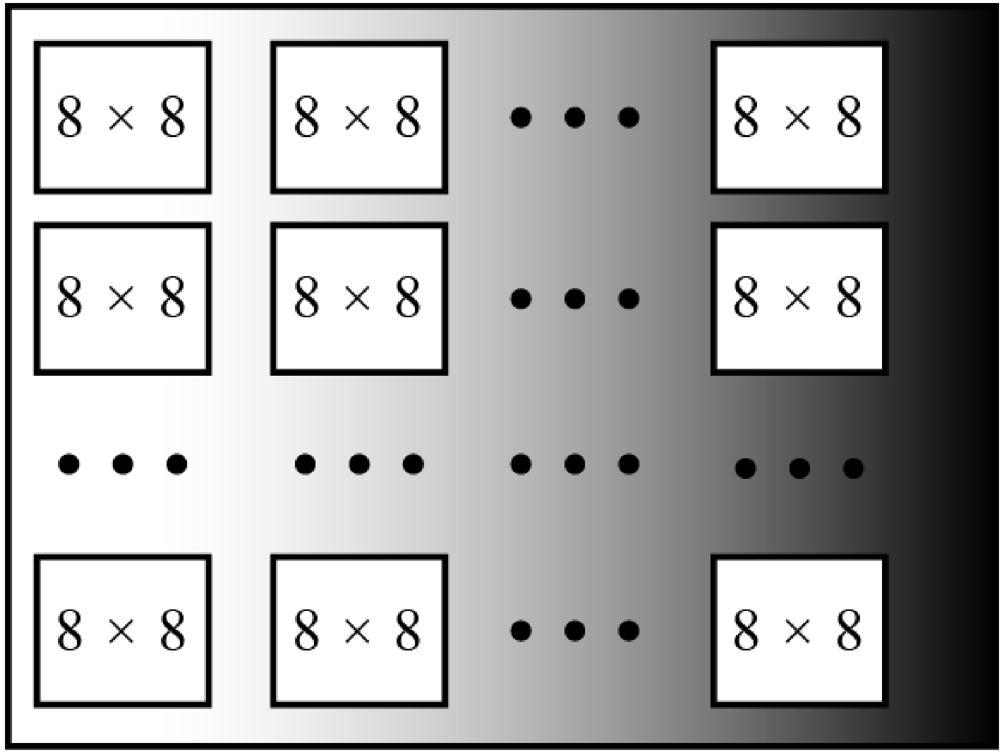


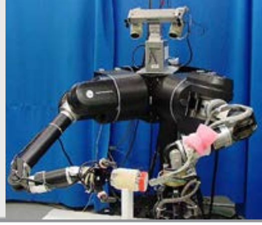
- Color space for JPEG compression
  - RGB values are a combination of brightness and color
  - If the RGB values are separated into a luminance and a color component, the color component is more compressed



JPEG Compression

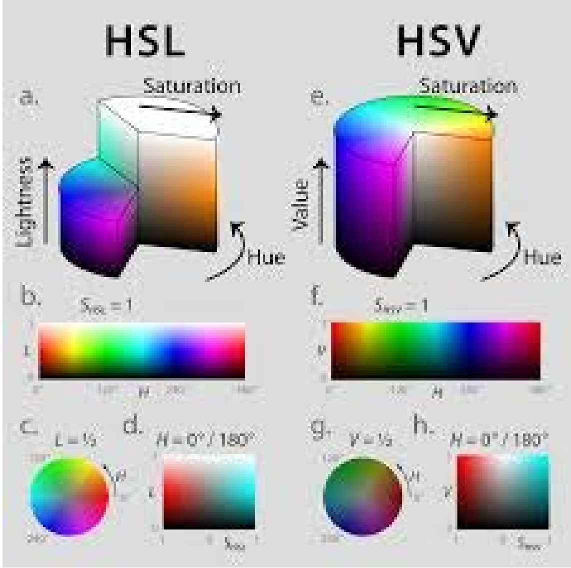
- Image is divided into blocks of 8 × 8 pixel blocks to decrease the number of calculations because the number of mathematical operations for each image is the square of the number of units.





JPEG Compression

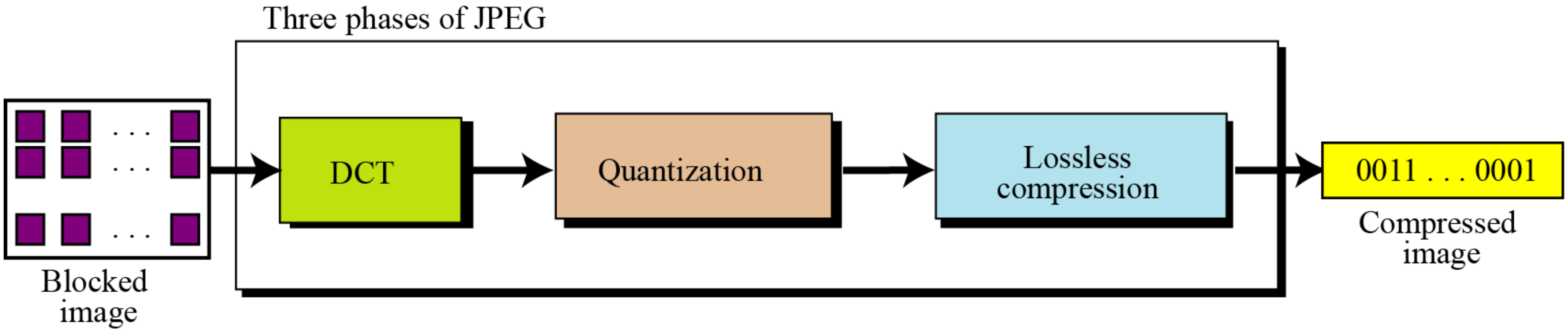
- Colorspace conversion and Downsampling:
  - Separation of the color components of the luminance information
  - Insensitivity of human eye to rapid color changes allows coarser sampling of the color components
    -> No loss of subjective quality
    -> Significant data reduction



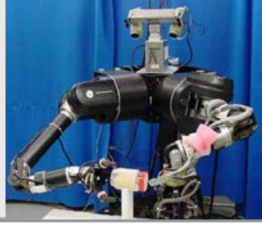
- DCT Quantization in spectral component
  - DCT for 8x8
  - Quantization of the 64 spectral coefficients by a quantization table (table determines quality of compressed image)

JPEG Compression

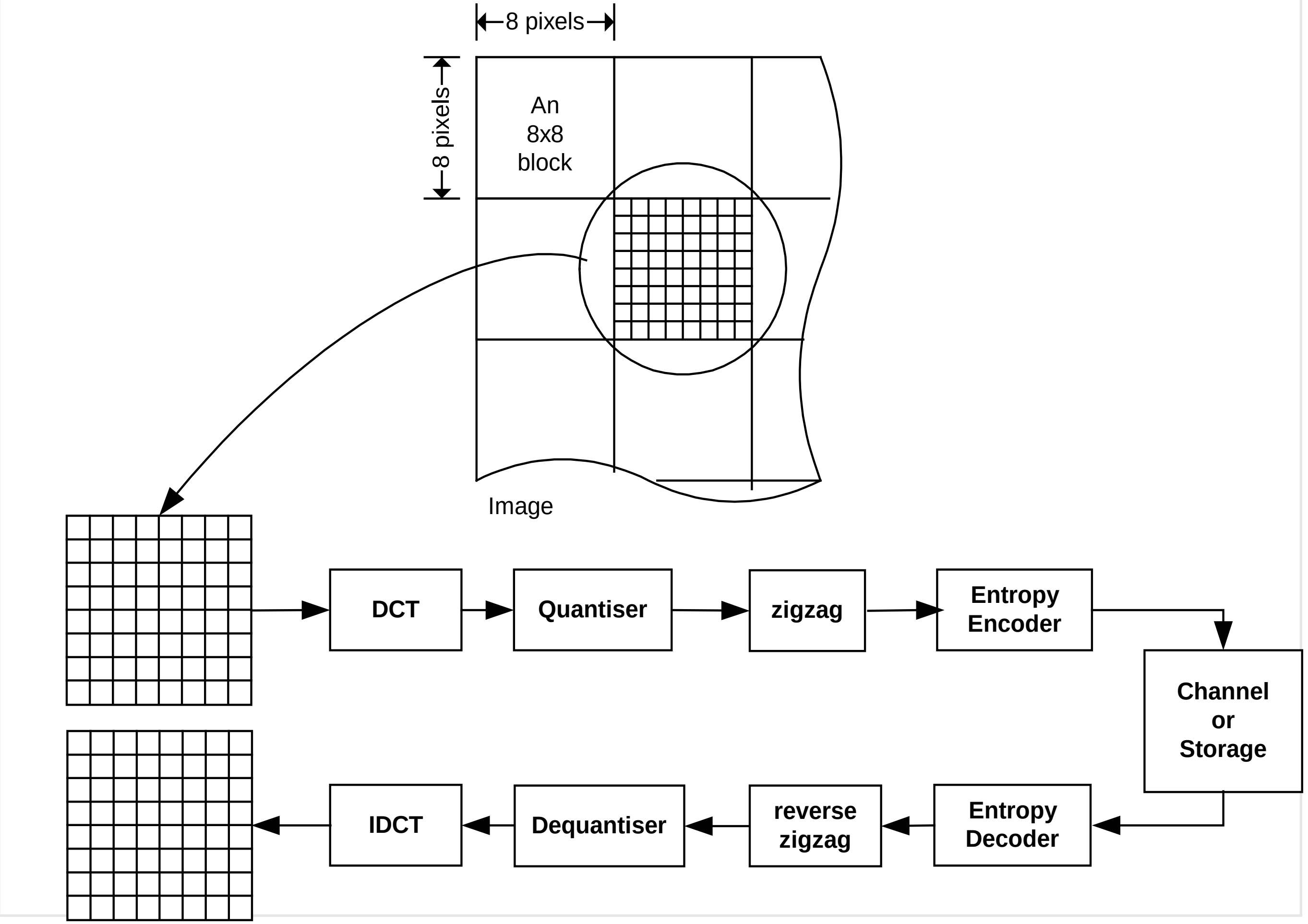
- Idea: change image into a linear (vector) set of numbers that reveals redundancies.
- Redundancies can be removed using one of the lossless compression methods



JPEG Compression

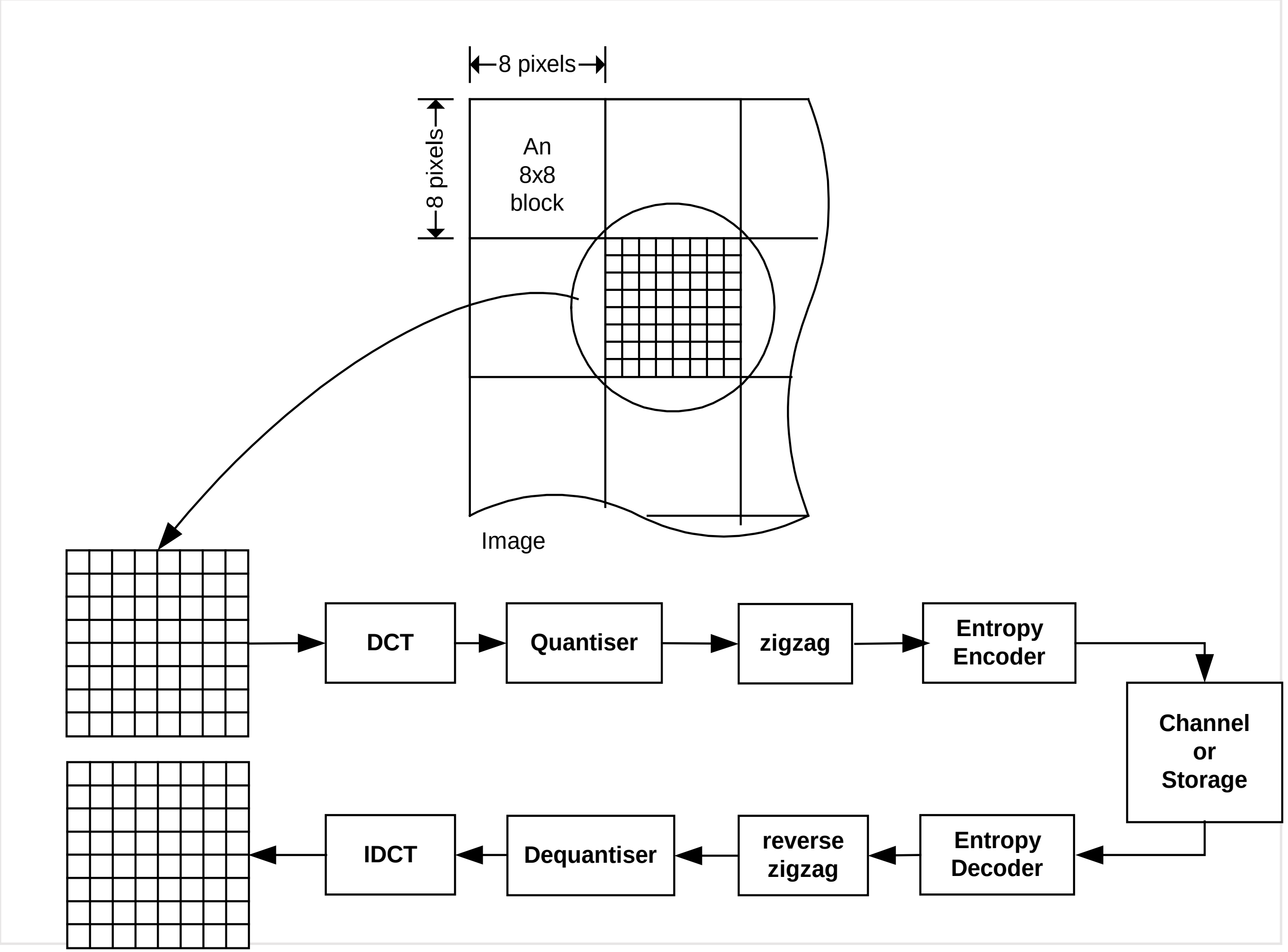


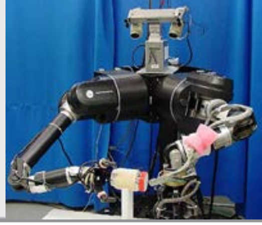
- To perform the JPEG coding, an image (in color or grey scales) is first subdivided into blocks of 8x8 pixels.
- The Discrete Cosine Transform (DCT) is then performed on each block.
- This generates 64 coefficients which are then quantized to reduce their magnitude.



JPEG Compression

- The coefficients are then reordered into a one-dimensional array in a zigzag manner before further entropy encoding.
- The compression is achieved in two stages; the first is during quantization and the second during the entropy coding process.
- JPEG decoding is the reverse process of coding.

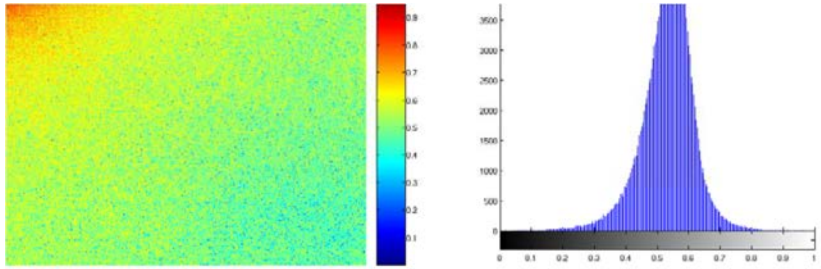




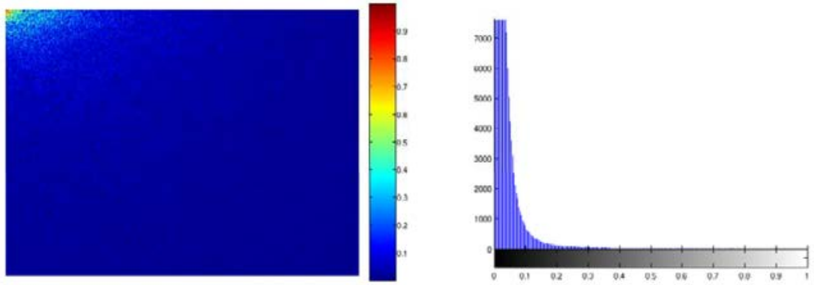
Discrete Cosine Transform

- Similar to Discrete Fourier Transform (DFT) but much better for energy compactation
  - Example: we see amplitude spectra of image under DFT and DCT
  - Note the much more concentrated histogram obtained with DCT

- Why is energy compaction important?
  - The main reason is image compression



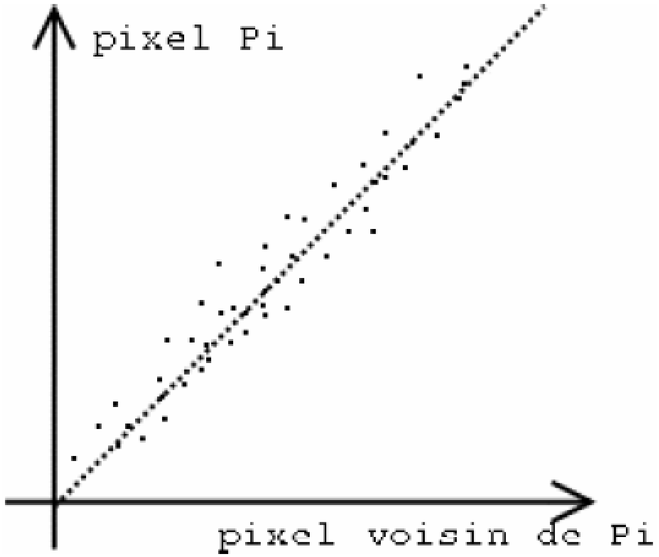
DFT



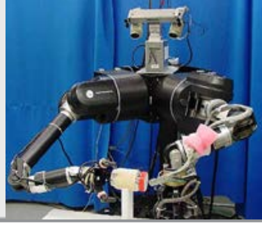
DCT

DCT

- The transform throws away correlations
  - If you make a plot of the value of a pixel as a function of one of its neighbors
  - You will see that the pixels are highly correlated (i.e. most of the time they are very similar)
  - This is just a consequence of the fact that surfaces are smooth



DCT



- DCT-based codecs use a two-dimensional version of the transform.
- The 2-D DCT of an 8 x 8 block:

$$F(u,v) = \sum_{x=0}^{7}\sum_{y=0}^{7} \alpha(u)\alpha(v) f(x,y) \cos[\frac{\pi}{8}(x+\frac{1}{2})u] \cos[\frac{\pi}{8}(y+\frac{1}{2})v]$$

u is the horizontal spatial frequency, for the integers $0 \le u < 8$

v is the vertical spatial frequency, for the integers $0 \le v < 8$

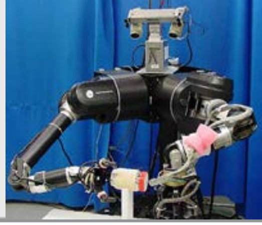
$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{8}}, & \text{if } u = 0 \\ \sqrt{\frac{2}{8}}, & \text{otherwise} \end{cases}$$ is a normalizing scale factor to make the transformation orthonormal

f(x,y) is the pixel value at coordinates ***(x,y)***

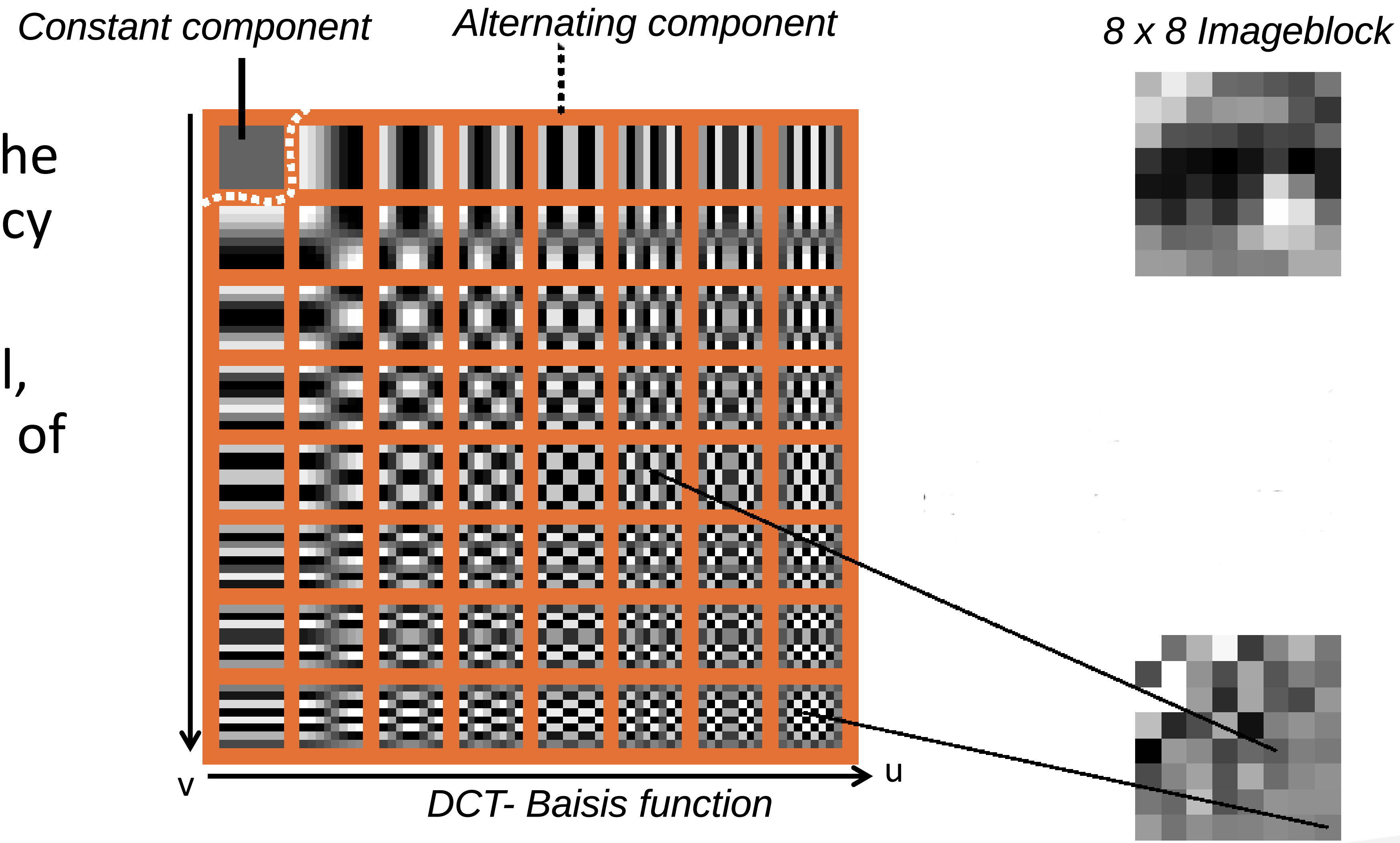
F(u,v) is the DCT coefficient at coordinates ***(u,v)***

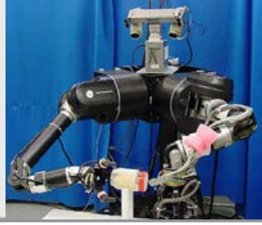
◆ **Note: The DCT decomposes a signal into a series of harmonic cosine functions.**

DCT Basis Functions

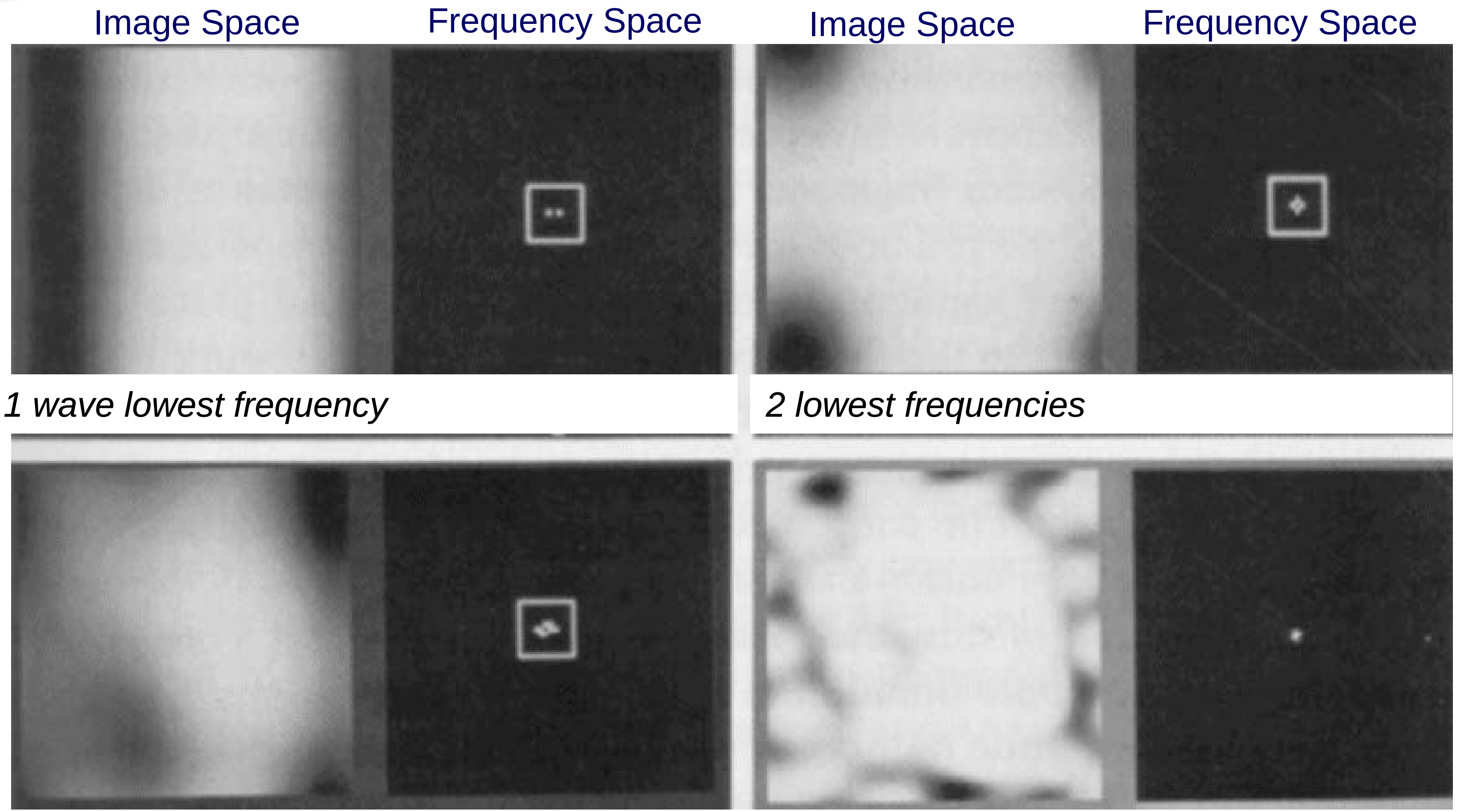


- DCT values indicate the weighting of frequency images
- The brighter the pixel, the greater the value of DCT





Decomposing into Frequencies



1 wave lowest frequency

2 lowest frequencies

4 lowest frequencies

16 lowest frequencies

(Example computed with DFT, which is similar to DCT)

Decomposing into Frequencies

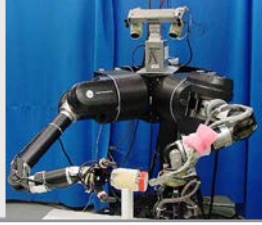
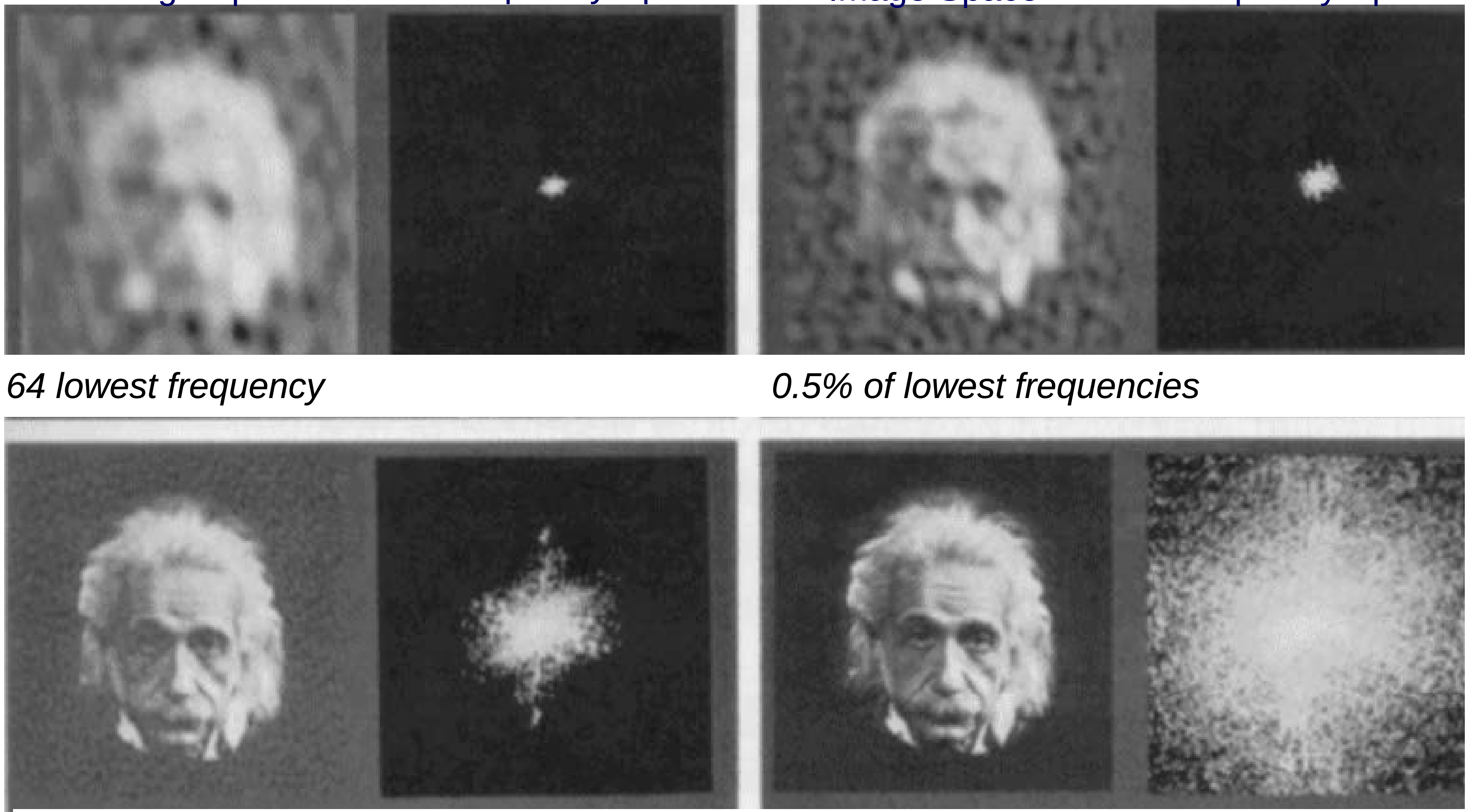


Image Space Frequency Space Image Space Frequency Space



64 lowest frequency

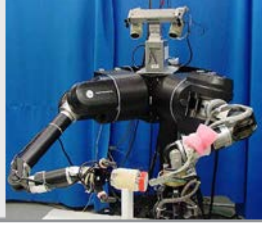
0.5% of lowest frequencies

20% of lowest frequencies

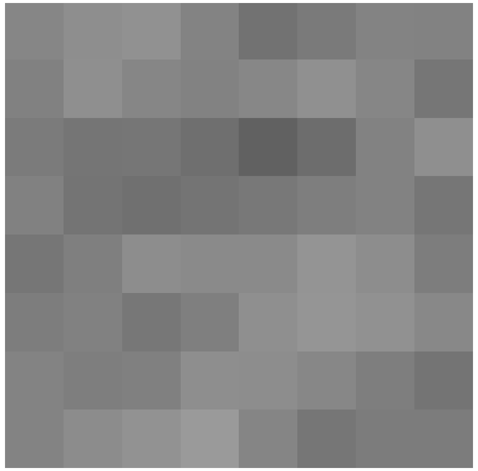
All frequencies

(Example computed with DFT, which is similar to DCT)

DCT



- DCT sorts values from the lowest to the highest frequency
- In image blocks it is likely that the energy is concentrated in low frequencies
  - Regions with smooth colors or little details (= low spatial frequencies) have high values
  - Regions with different colors and detail (= high spatial frequencies), most values are almost zero.



8x8 Block (Luminance)

134	142	145	131	114	122	131	130
129	143	134	130	135	144	134	118
123	117	118	111	97	109	130	143
129	116	112	116	120	126	130	118
118	127	141	138	138	148	141	125
125	129	119	127	143	149	145	136
131	126	128	142	141	135	126	116
131	140	146	154	133	118	124	124

Pixelvalues

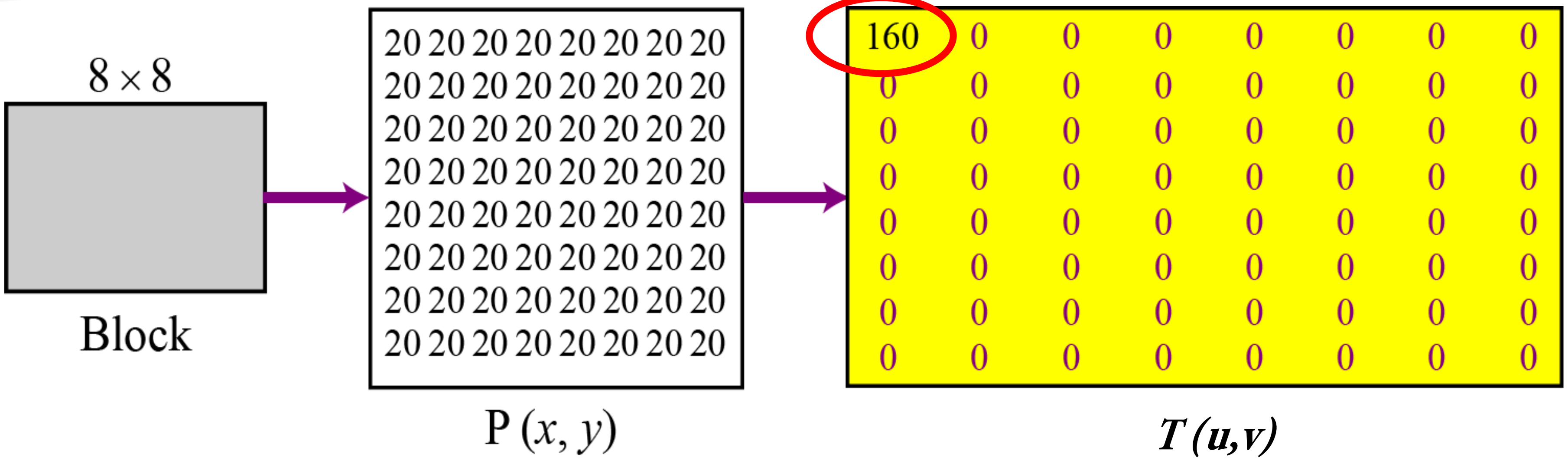
Low frequencies

1037	-1	-6	1	-12	8	-4	-4
-16	1	28	-6	-14	0	4	0
19	32	-7	-19	2	-1	-4	-3
29	-9	-14	13	-10	-6	1	0
4	14	-6	-13	-2	7	1	2
-26	2	16	2	11	6	1	1
-10	-11	27	-18	4	1	0	0
-2	1	1	-19	-1	6	6	0

High frequencies

DCT- Values

DCT Examples

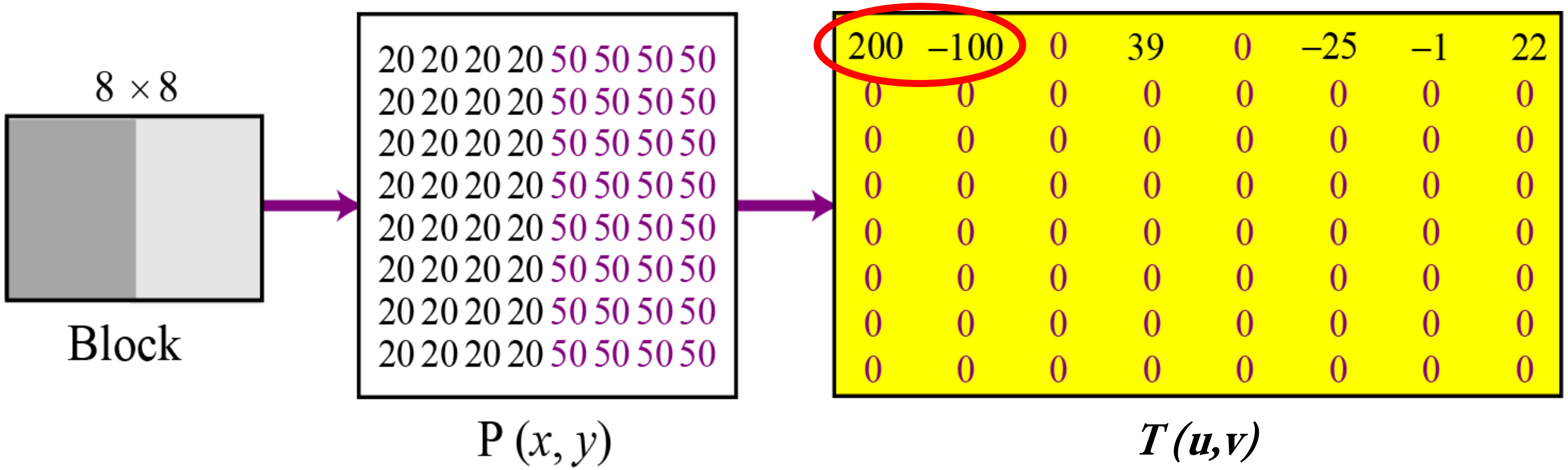


P (x, y)

20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20

T (u,v)

160	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



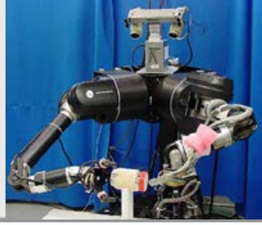
P (x, y)

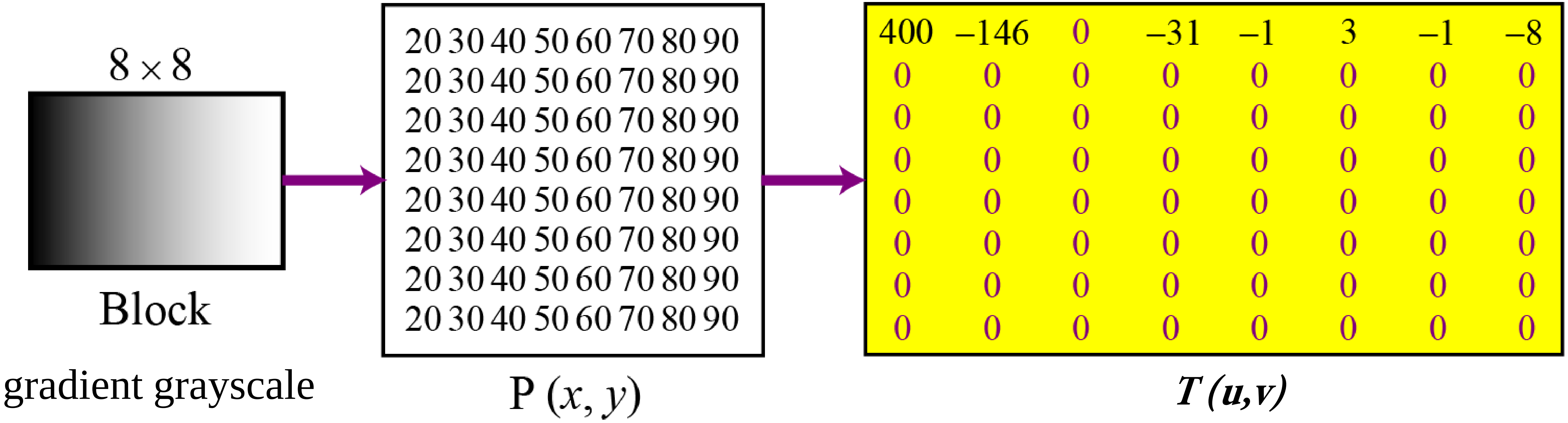
20	20	20	20	50	50	50	50
20	20	20	20	50	50	50	50
20	20	20	20	50	50	50	50
20	20	20	20	50	50	50	50
20	20	20	20	50	50	50	50
20	20	20	20	50	50	50	50
20	20	20	20	50	50	50	50
20	20	20	20	50	50	50	50

T (u,v)

200	–100	0	39	0	–25	–1	22
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

DCT Examples





8×8

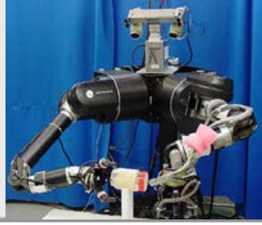
Block

gradient grayscale

$$P(x,y) = \begin{matrix} 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 \\ 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 \\ 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 \\ 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 \\ 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 \\ 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 \\ 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 \\ 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 \end{matrix}$$

$$T(u,v) = \begin{matrix} 400 & -146 & 0 & -31 & -1 & 3 & -1 & -8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Quantization



- Compression effect: DCT values are approximated (quantized) to make them smaller and to repeat themselves
  - Each DCT value is divided by a quantization factor and then rounded
  - The larger the quantization factor, the smaller the values to be stored

134	142	145	131	114	122	131	130
129	143	134	130	135	144	134	118
123	117	118	111	97	109	130	143
129	116	112	116	120	126	130	118
118	127	141	138	138	148	141	125
125	129	119	127	143	149	145	136
131	126	128	142	141	135	126	116
131	140	146	154	133	118	124	124

Pixelvalues

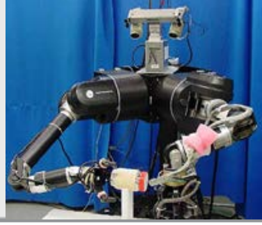
1037	-1	-6	1	-12	8	-4	-4
-16	1	28	-6	-14	0	4	0
19	32	-7	-19	2	-1	-4	-3
29	-9	-14	13	-10	-6	1	0
4	14	-6	-13	-2	7	1	2
-26	2	16	2	11	6	1	1
-10	-11	27	-18	4	1	0	0
-2	1	1	-19	-1	6	6	0

DCT- Values

130	0	0	0	-1	0	0	0
-2	0	3	0	-1	0	0	0
2	3	0	-1	0	0	0	0
3	-1	-1	1	0	0	0	0
0	1	0	-1	0	0	0	0
-2	0	2	0	0	0	0	0
-1	-1	2	-1	0	0	0	0
0	0	0	-1	0	0	0	0

Quantized DCT- Values

Quantization Matrix



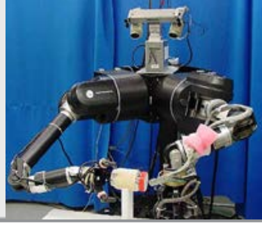
- Quantization factor for each DCT value is defined using a quantization matrix
  - Disturbances in low frequency parts of the image are perceived strongly
  - Disturbances in high frequency parts of the image are less noticeable
  - Quantization matrices in JPEG standard ensures that the DCT values for low frequencies are stored more accurately than for high frequencies
  - Quantization table can be freely selected in the compression

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

Quantization Matrix

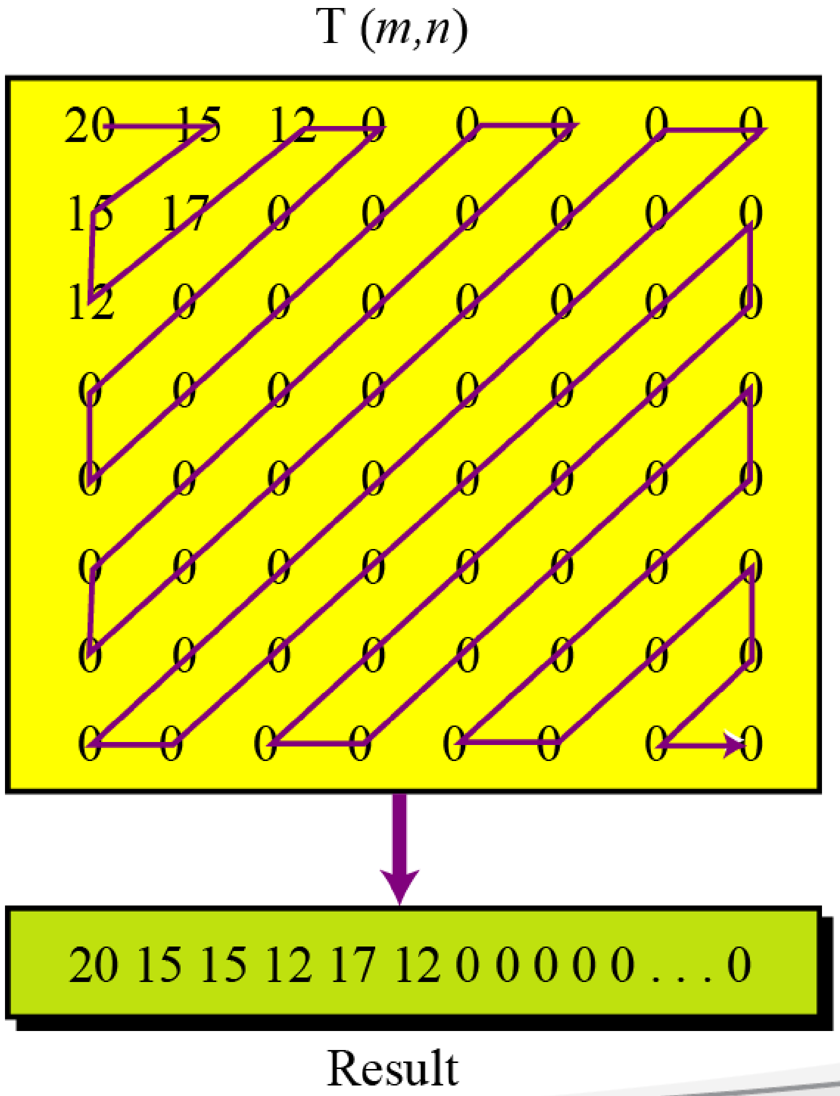
$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Quantization Matrix



Compression

- After quantization values are read from the table, and redundant 0s are removed.
- To cluster 0s together process reads the table diagonally in a zigzag fashion because if image does not have fine changes bottom right corner of table is all 0s.
- JPEG uses run-length encoding at the compression phase to compress the bit pattern resulting from the zigzag linearization.



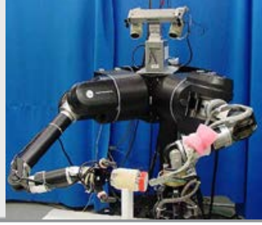
T (m,n)

20	15	12	0	0	0	0	0
15	17	0	0	0	0	0	0
12	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

20 15 15 12 17 12 0 0 0 0 0 . . . 0

Result

Quantization



Coding

134	142	145	131	114	122	131	130
129	143	134	130	135	144	134	118
123	117	118	111	97	109	130	143
129	116	112	116	120	126	130	118
118	127	141	138	138	148	141	125
125	129	119	127	143	149	145	136
131	126	128	142	141	135	126	116
131	140	146	154	133	118	124	124

Pixelvalues

1037	-1	-6	1	-12	8	-4	-4
-16	1	28	-6	-14	0	4	0
19	32	-7	-19	2	-1	-4	-3
29	-9	-14	13	-10	-6	1	0
4	14	-6	-13	-2	7	1	2
-26	2	16	2	11	6	1	1
-10	-11	27	-18	4	1	0	0
-2	1	1	-19	-1	6	6	0

DCT- Values

130	0	0	0	-1	0	0	0
-2	0	3	0	-1	0	0	0
2	3	0	-1	0	0	0	0
3	-1	-1	1	0	0	0	0
0	1	0	-1	0	0	0	0
-2	0	2	0	0	0	0	0
-1	-1	2	-1	0	0	0	0
0	0	0	-1	0	0	0	0

Quantized DCT- Values

Decoding

1040	0	0	0	-9	0	0	0
-12	0	24	0	-10	0	0	0
14	24	0	-10	0	0	0	0
24	-8	-9	10	0	0	0	0
0	9	0	-10	0	0	0	0
-19	0	10	0	0	0	0	0
-9	-10	21	-12	0	0	0	0
0	0	0	-14	0	0	0	0

Reconstructed DCT- Values

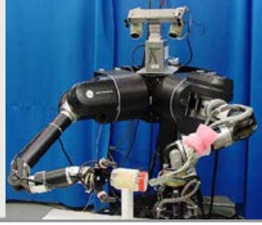
136	141	138	125	119	125	132	134
136	137	133	130	134	139	133	121
121	125	122	112	107	117	130	138
125	123	119	117	121	128	127	122
123	129	135	139	140	139	136	131
129	125	124	130	139	144	141	136
129	130	134	139	140	135	127	120
132	138	144	141	131	122	122	127

Reconstructed Pixels (IDCT)

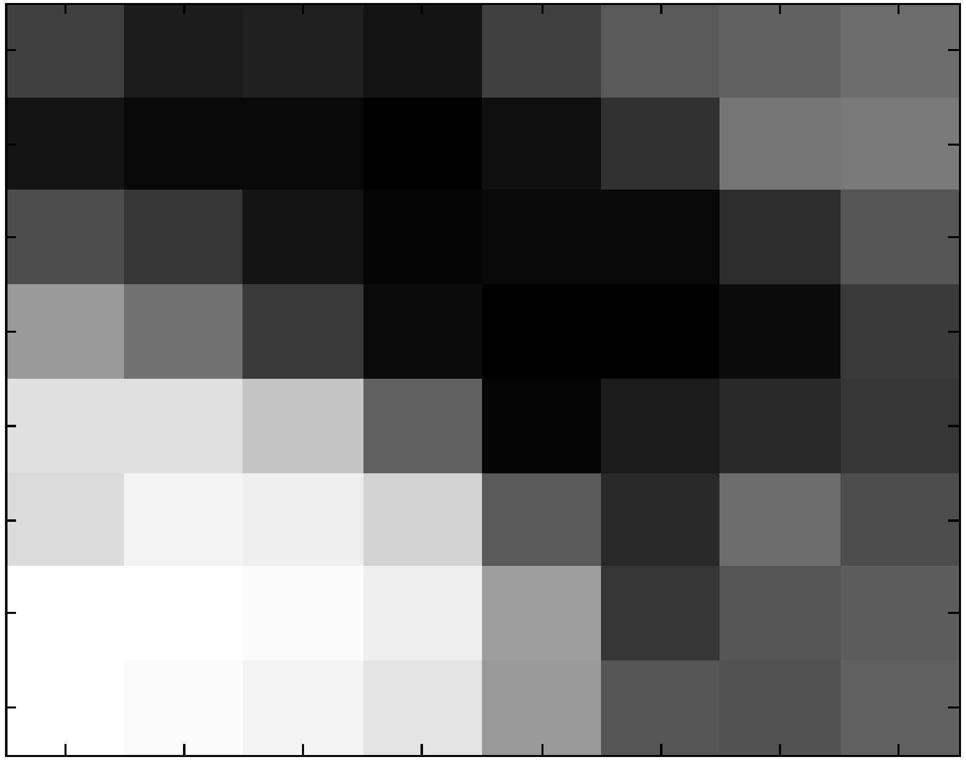
-2	1	7	6	-5	-3	-1	-4
-7	6	1	0	1	5	1	-3
2	-8	-4	-1	-10	-8	0	5
4	-7	-7	-1	-1	-2	3	-4
-5	-2	6	-1	-2	9	5	-6
-4	4	-5	-3	4	5	4	0
2	-4	-6	3	1	0	-1	-4
1	2	2	13	2	-4	1	-3

Difference

Can You Tell the Difference?



Original



Compressed

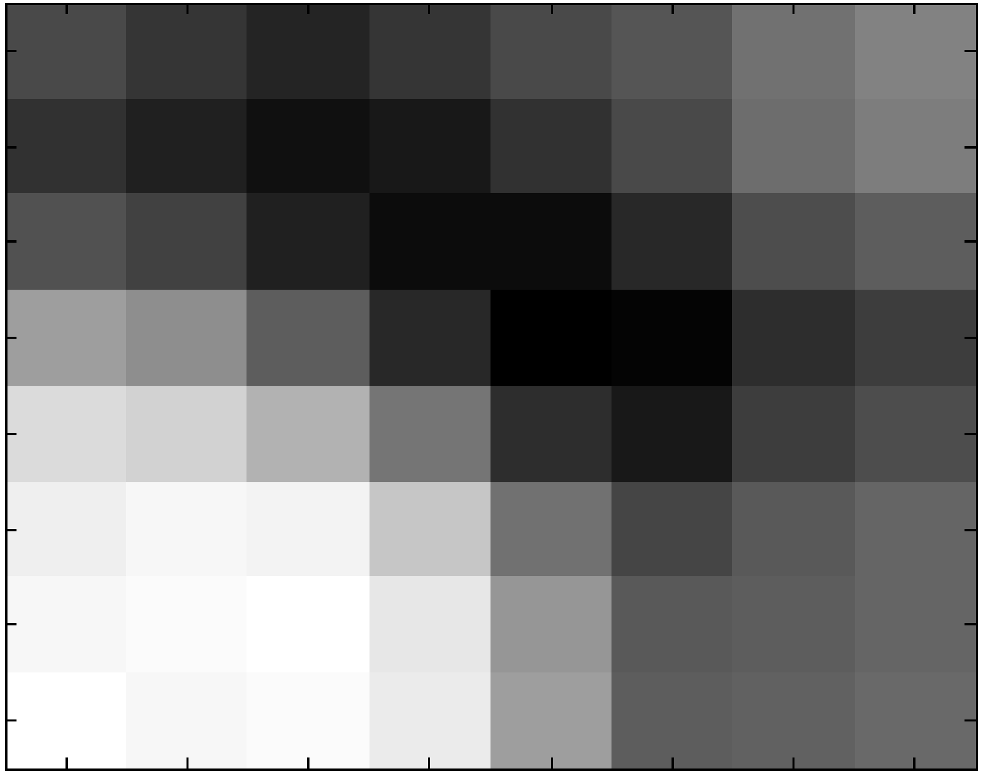
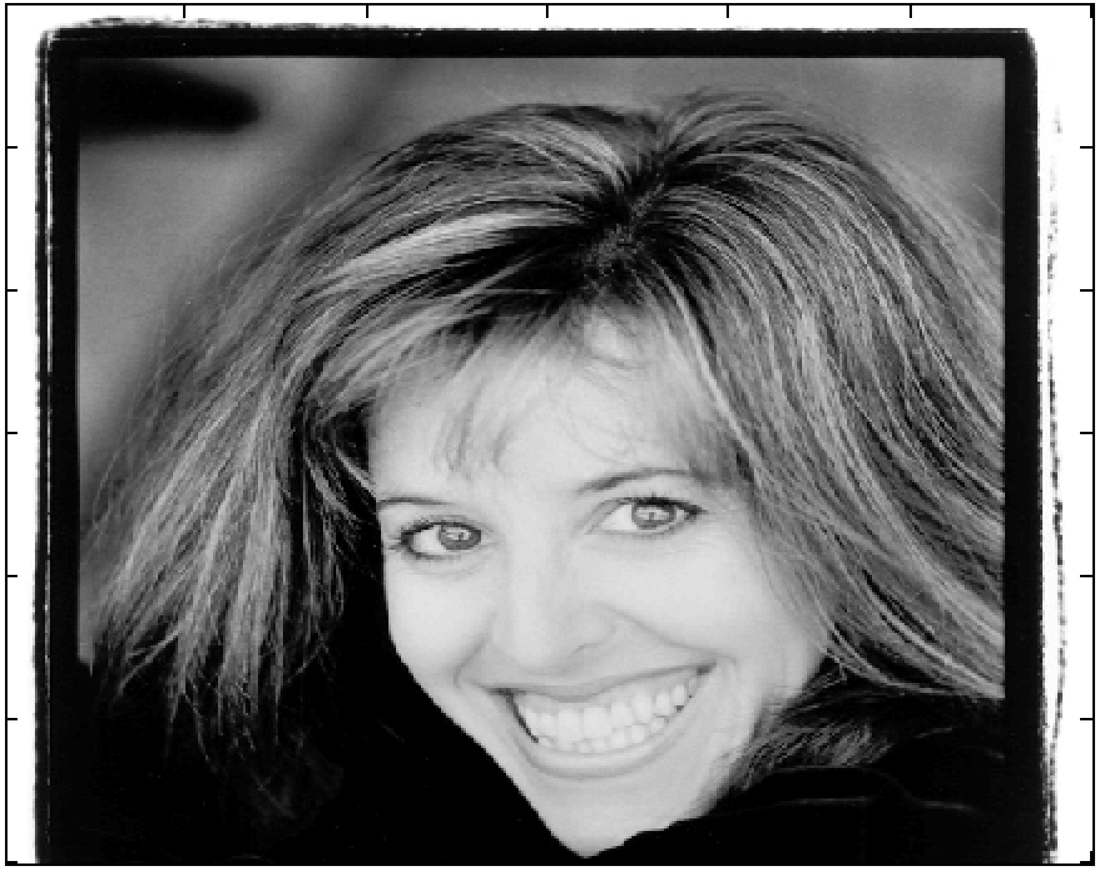
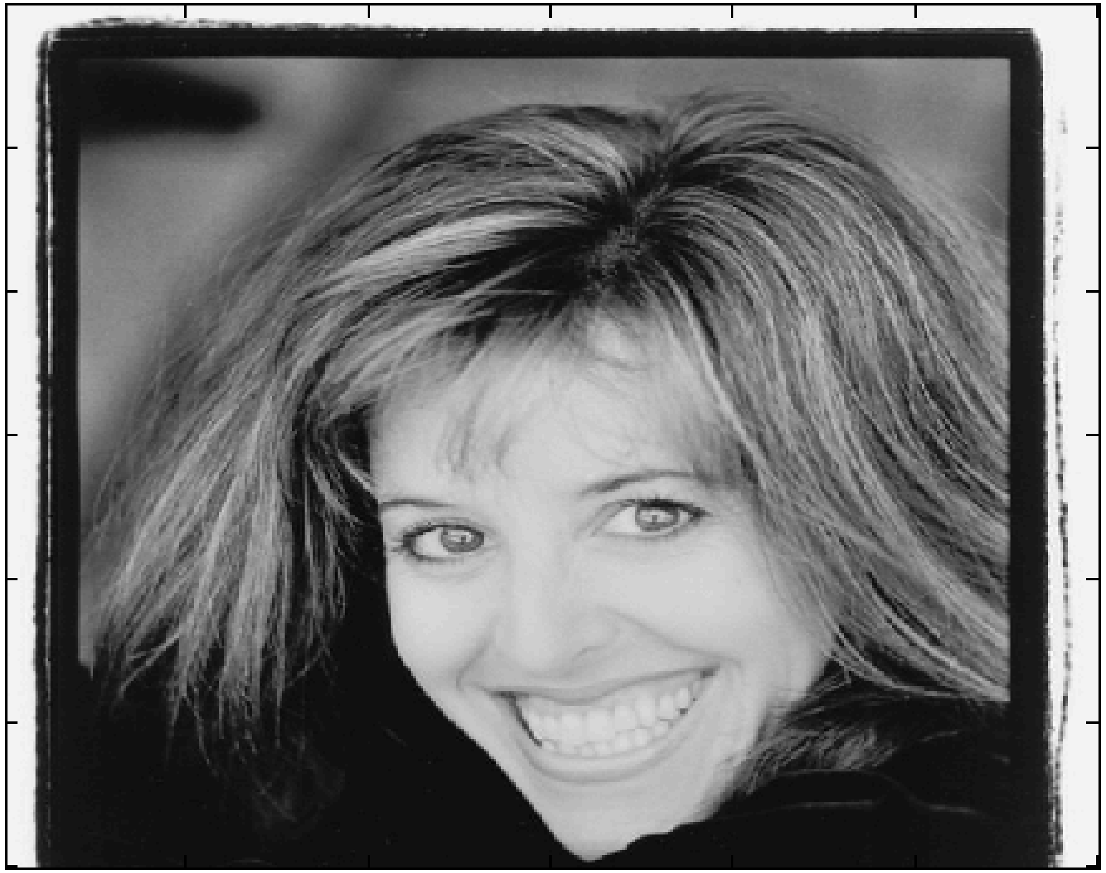


Image Compression

Original

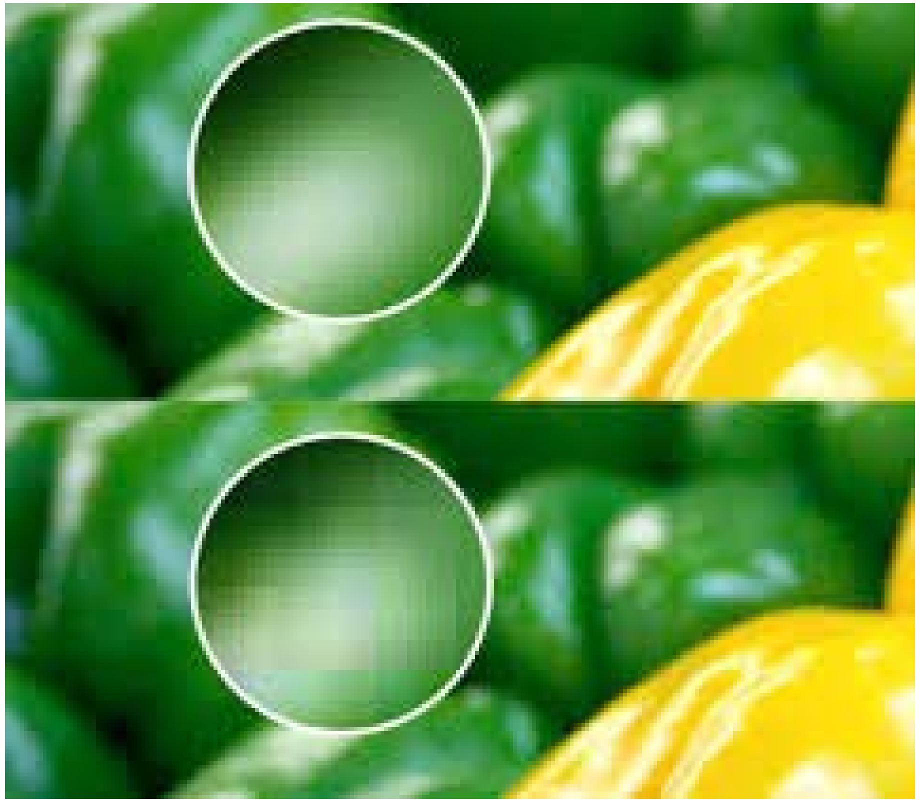


Compressed

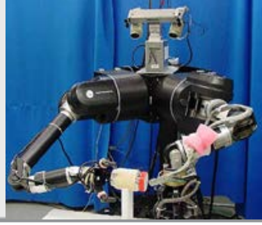


JPEG Properties

- Weakness:
  - Behavior in sharp transitions (e.g. fonts)
  - Emergence of the 8x8 blocks at high compression rates.
- JPEG compression can reduce an image to a fifth of its original size (without visual impairment)
- The greater the compression (quantization), the more artefacts occur (block formation)
- JPEG is made for natural images and not for artificial images (computer graphics)!



Video Compression



Evolution of Video Media

- Film
  - Invented in late 18th century, still used today

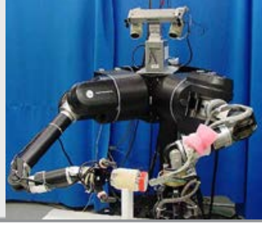
- VHS
  - Released in 1976, rapidly disappearing

Evolution of Video Media

- DVD
  - Released in 1996, dominant for over a decade

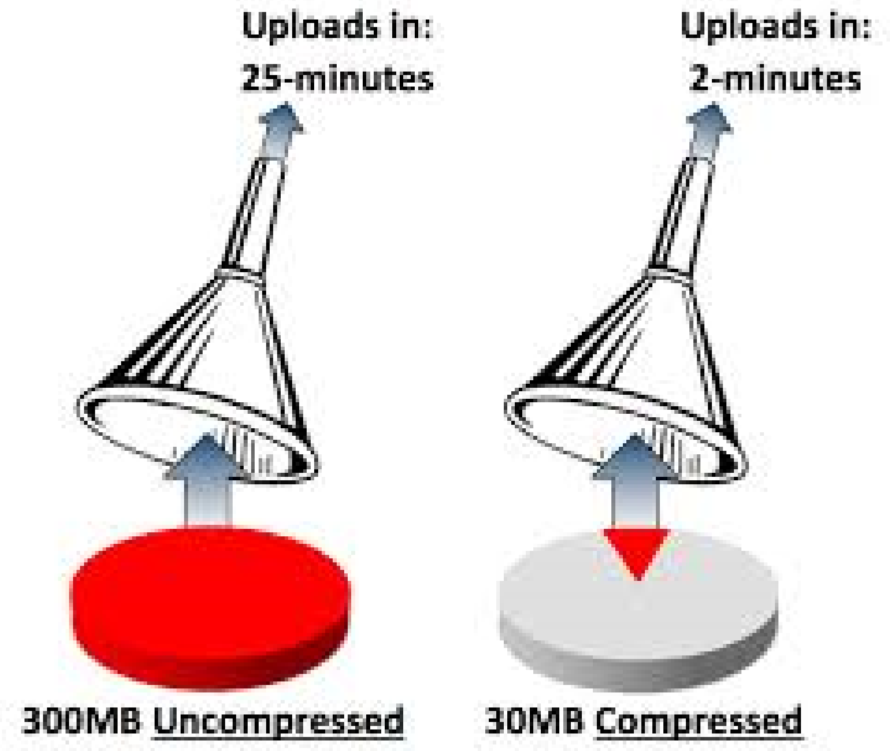
- Hard Disk
  - Around for many years, only recently widely used for storing video (helped by explosion of Internet)

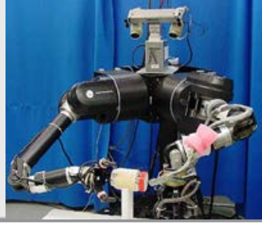
Videocompression



- Single Image
  - Size 720 x 576 px
  - Pixel resolution: 1 Byte/RGB Value
    → 720 x 576 x 3 (Byte) ~ 1.215 KB

- Image Sequence
  - 25 fps
    → 720 x 576 x 25 x 3 (Byte) ~ 30.375 KB/s

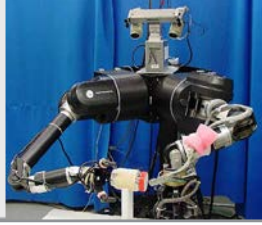




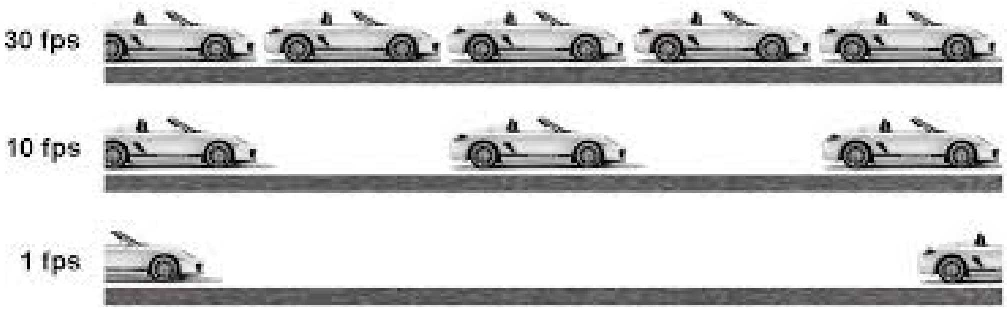
TMI! (Too Much Information)

- Unlike image encoding, video encoding is rarely done in lossless form
- No storage medium has enough capacity to store a practical sized lossless video file
  - Lossless DVD video - 221 Mbps
  - Compressed DVD video - 4 Mbps
  - 50:1 compression ratio!

Definitions

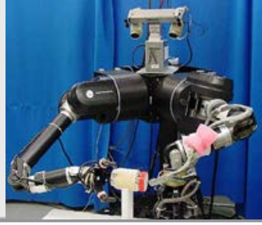


- Bitrate
  - Information stored/transmitted per unit time
  - Usually measured in Mbps (Megabits per second)
  - Ranges from < 1 Mbps to > 40 Mbps
- Resolution
  - Number of pixels per frame
  - Ranges from 160x120 to 1920x1080

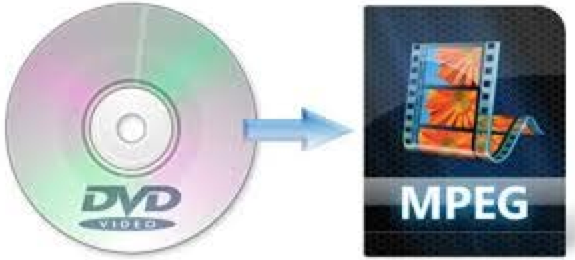


- FPS (frames per second)
  - Usually 24 (cinema), 25 (PALi, HDTVi), 30 (NTSCi), or 50,60 (DVD, HDTVp)
  - Don’t need more because of limitations of the human eye (16)

MPEG (Moving Pictures Expert Group)

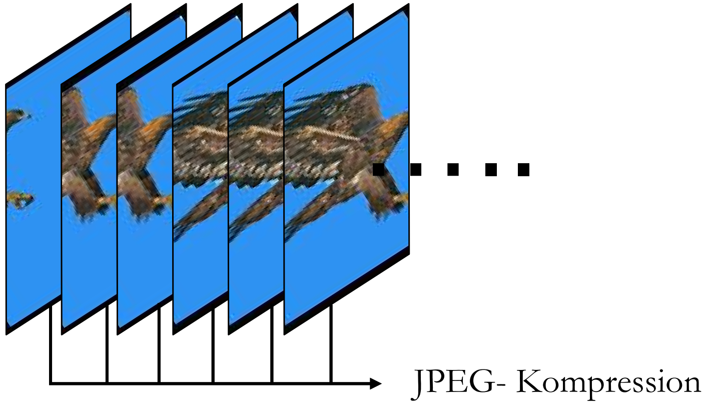


- Committee of experts that develops video encoding standards
- Until recently, was the only game in town (still the most popular, by far)
- Suitable for wide range of videos
  - Low resolution to high resolution
  - Slow movement to fast action
- Can be implemented either in software or hardware



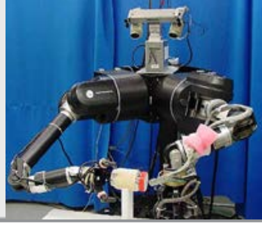
M- JPEG

- Videosequenzes
- Single image compression using JPEG

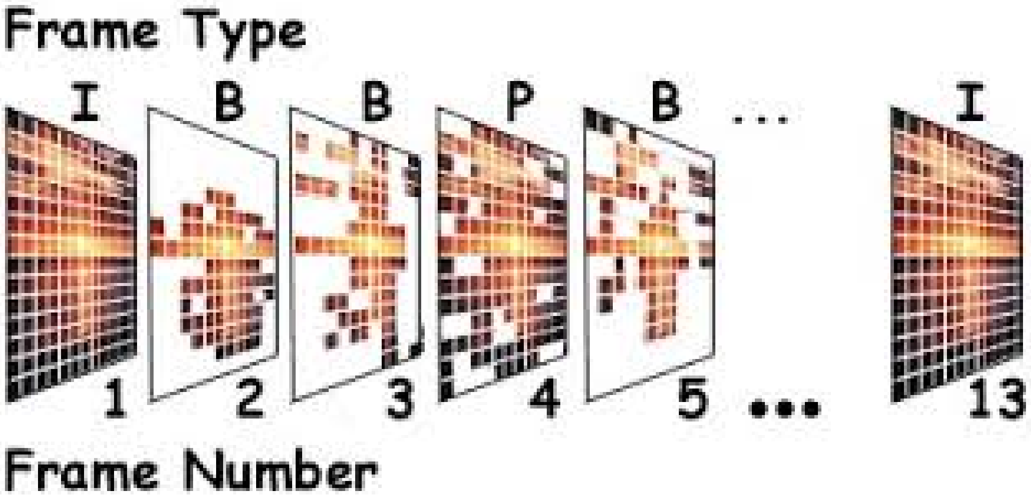


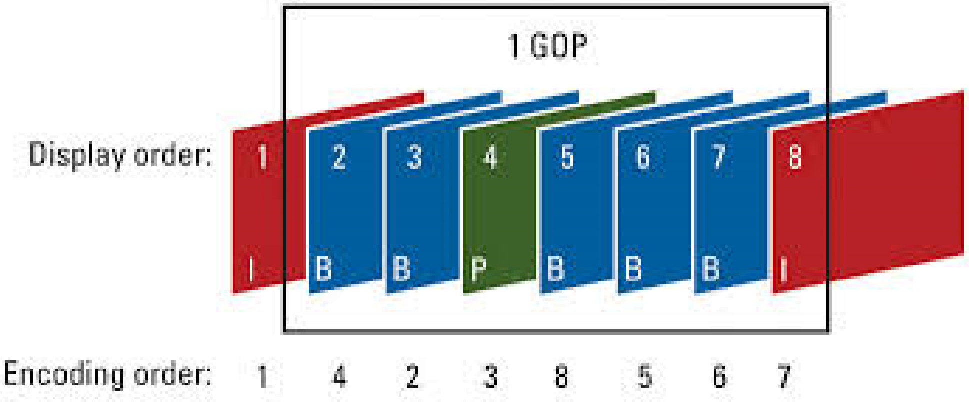
Benefits	Drawbacks
Constant Image Quality	Fluctuating bandwidth / frame rate
Fast computation	High memory requirements
Robust with respect to packet loss	No Audio

Types of Frames

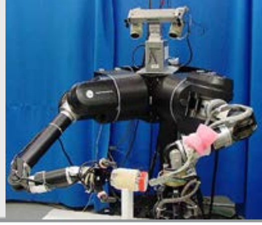


- I frame (intra-coded)
    - Coded without reference to other frames
- P frame (predictive-coded)
    - Coded with reference to a previous reference frame (either I or P)
    - Size is usually about 1/3rd of an I frame
- B frame (bi-directional predictive-coded)
    - Coded with reference to both previous and future reference frames (either I or P)
    - Size is usually about 1/6th of an I frame

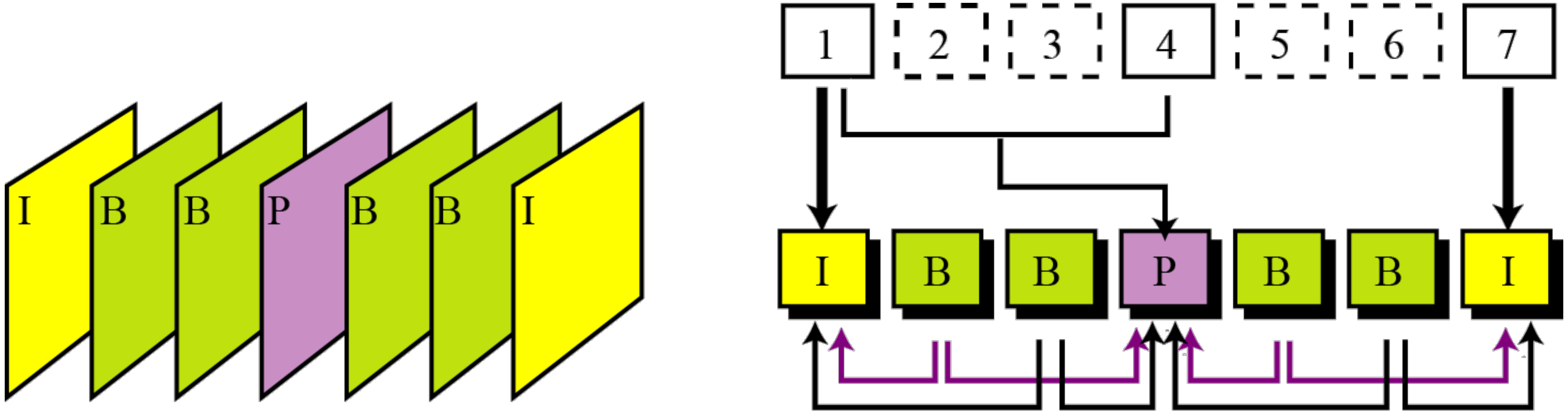




GOP (Group of Pictures)

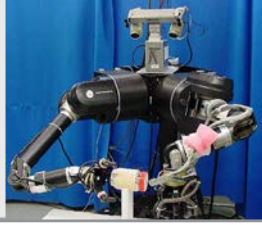


- GOP is a set of consecutive frames that can be decoded without any other reference frames
- Usually 12 or 15 frames
- Transmitted sequence is not the same as displayed sequence
- Random access to middle of stream – Start with I frame



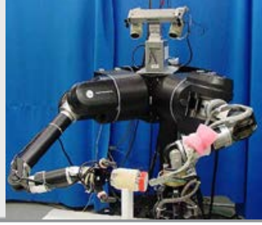
a. Frames b. Frame construction

Evolution of MPEG



- MPEG-1 (1992)
  - Initial audio/video compression standard
  - Used by VCD’s
  - MP3 = MPEG-1 audio layer 3
  - Target of 1.5 Mb/s bitrate at 352x240 resolution
  - Only supports progressive pictures

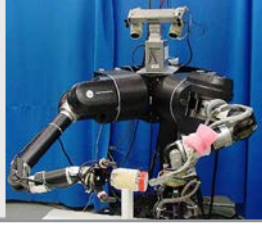
Evolution of MPEG



- MPEG-2 (1994)
  - Widely used in DVD and Digital TV
  - Ubiquity in hardware implies that it will be here for a long time
    - Transition to HDTV has taken over 10 years and is not finished yet
  - Different profiles and levels allow for quality control

- MPEG-3
  - Originally developed for HDTV, but abandoned when MPEG-2 was determined to be sufficient

Evolution of MPEG



- MPEG-4 (1998)
    - Includes support for AV “objects”, 3D content, low bitrate encoding, and DRM
    - In practice, provides equal quality to MPEG-2 at a lower bitrate, but often fails to deliver outright better quality
    - MPEG-4 Part 10 is H.264, which is used in HD-DVD and Blu-Ray
    - DivX, Xvid, HDX4, QuickTime

- MPEG-7 (2002)
    - Multimedia content description standard
    - Digital library